NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

S65922

INVESTIGATION OF INCOMPRESSIBLE CASCADE FLOWS USING A VISCOUS/INVISCID INTERACTIVE CODE

by

Zeev Snir

December 1988

Thesis Advisor M. F. Platzer

Approved for public release; distribution is unlimited.

T242366

REPORT DOCUMENTATION PAGE

Report Security Classification Unclassified		1b Restrictive Markings	
Security Classification Authority		3 Distribution Availability of Report Approved for public release; distribution is unlimited.	
Declassification Downgrading Schedule			
Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
Name of Performing Organization aval Postgraduate School	6b Office Symbol *(if applicable)* 67	7a Name of Monitoring Organization Naval Postgraduate School	
Address *(city, state, and ZIP code)* onterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	
Name of Funding Sponsoring Organization	8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number	
Address *(city, state, and ZIP code)*		10 Source of Funding Numbers	
		Program Element No / Project No / Task No / Work Unit Accession No	

Title *(include security classification)* INVESTIGATION OF INCOMPRESSIBLE CASCADE FLOWS USING A ISCOUS/INVISCID INTERACTIVE CODE

Personal Author(s) Zeev Snir

a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
aster's Thesis	From To	December 1988	135

Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or po-ion of the Department of Defense or the U.S. Government.

Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
eld	Group	Subgroup	thesis, cascade flow, viscous inviscid interaction code.

Abstract *(continue on reverse if necessary and identify by block number)*

A two dimensional, incompressible viscous inviscid interaction computer code, designed to compute cascade flows, was vestigated. Comparison of the flow characteristics predicted by the code with experimentally available data indicates that e code predicts reasonably well flow parameters on lightly loaded cascades. However, the code fails to predict correctly the tual boundary layer development and the velocity distribution for highly loaded cascades. It is concluded that further im-ovement of the code is needed and recommendations are presented to achieve the required improvements.

Distribution Availability of Abstract unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified	
a Name of Responsible Individual laximilian F. Platzer	22b Telephone *(include Area code)* (408) 646-2311	22c Office Symbol 67Pl

D FORM 1473,84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Investigation of Incompressible Cascade Flows Using a Viscous/Inviscid Interactive Code

by

Zeev Snir
Major, Israeli Air Force
B.S., Technion, Haifa, Israel, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1988

ABSTRACT

A two dimensional, incompressible viscous inviscid interaction computer code, designed to compute cascade flows, was investigated. Comparison of the flow characteristics predicted by the code with experimentally available data indicates that the code predicts reasonably well flow parameters on lightly loaded cascades. However, the code fails to predict correctly the actual boundary layer development and the velocity distribution for highly loaded cascades. It is concluded that further improvement of the code is needed and recommendations are presented to achieve the required improvements.

Thesis
S65922
C.1

TABLE OF CONTENTS

I. INTRODUCTION 1

II. CASCADE FLOW PROBLEM FORMULATION. 2
A. INVISCID FLOW METHOD. 2
B. VISCOUS FLOW METHOD. 3
1. Boundary Layer Theory 3
2. Interactive Boundary Layer Method 6
3. Interactive Model 8
4. Turbulence Model. 9
5. Transition 12

III. DESCRIPTION OF THE COMPUTER CODE 13
A. GENERAL STRUCTURE OF THE MAIN PROGRAM 13
B. DESCRIPTION OF THE SUBROUTINES 13
1. Subroutine POTNL 13
2. Subroutine CASBLP 14
3. Subroutine COMPBL 14
4. Subroutine BL2D 14
5. Subroutine MAIN2 15
6. Subroutine OUTPUT 15
7. Subroutine TRANS 16
8. Subroutine FILLUP 16
9. Subroutine EDDY 17
10. Subroutine INTL 17

IV. RESULTS AND DISCUSSION 18
A. CD CASCADE 19
1. Transition location and intermittency distribution. 19
2. External Velocity Distribution 30
3. Boundary Layer Thickness 34
4. Comparison to a Navier Stokes Code. 42

B. P & W CASCADE 46
C. C4 CASCADE 51
1. Displacement Thickness 51
2. External Velocity and Velocity Profiles Comparisons 57

V. CONCLUSIONS AND RECOMMENDATIONS 62
A. CONCLUSIONS 62
B. RECOMMENDATIONS 62

APPENDIX A. COMPUTER CODE LISTING 64

APPENDIX B. C4 CASCADE 119
A. EXPERIMENTAL RESULTS 119
B. C4 CASCADE COORDINATES 121

LIST OF REFERENCES 123

INITIAL DISTRIBUTION LIST 125

LIST OF FIGURES

Figure 1. Controlled Diffusion cascade 20
Figure 2. Shape factor comparison on the upper surface 21
Figure 3. Shape factor comparison on the lower surface 22
Figure 4. Shape factor on the upper surface without velocity smoothing. 24
Figure 5. Displacement thickness 25
Figure 6. The effect of the intermittency model 26
Figure 7. Shape factor on the lower surface with transition input at 21%. 27
Figure 8. Shape factor at β = 46° on the upper surface 28
Figure 9. Shape factor at β = 46° on the lower surface. 29
Figure 10. External velocity at β = 40° 31
Figure 11. External velocity at β = 43.4° 32
Figure 12. External velocity at β = 46° 33
Figure 13. Displacement thickness on the lower surface (β = 40°) 35
Figure 14. Displacement thickness on the lower surface (β = 43.4°) 36
Figure 15. Displacement thickness on the lower surface (β = 46°) 37
Figure 16. Displacement thickness on the upper surface (β = 40°) 38
Figure 17. Displacement thickness on the upper surface (β = 43.4°) 39
Figure 18. Displacement thickness on the upper surface (β = 46°) 40
Figure 19. The effect of sharp trailing edge. 41
Figure 20. The results of the N. S. code at β = 40° 43
Figure 21. The results of the N. S. code at β = 43.4° 44
Figure 22. The results of the N. S. code at β = 46° 45
Figure 23. Pratt & Whitney cascade 47
Figure 24. Comparison of pressure coefficient. 48
Figure 25. Displacement thickness comparison 49
Figure 26. Velocity profile at 96.8% chord on the upper surface. 50
Figure 27. C4 Cascade 52
Figure 28. C4 cascade at β = 34.1° 53
Figure 29. C4 cascade at β = 36.3° 54
Figure 30. C4 cascade at β = 45.6° 55
Figure 31. C4 cascade at β = 47.7° 56

Figure 32. C4 cascade at β = 45.6° 58
Figure 33. C4 cascade at β = 47.7° 59
Figure 34. C4 cascade at β = 34.1° 60
Figure 35. C4 cascade at β = 36.3° 61

ACKNOWLEDGEMENTS

My sincere appreciation to Professor M. F. Platzer for his guidance and professional advice in conducting this thesis. I would also like to express my appreciation to Mr. A. Krainer for his help in running the computer code and for his constructive suggestions during the conduct of this thesis.

I. INTRODUCTION

The need for better and more efficient gas turbines requires the availability of cheap and reliable design tools for blades used in compressors and turbines. Computational methods are the preferred choice for such a design tool, considering the cost and complexity of wind tunnel experiments.

Among the computational methods available today, the logical choice seems to be a computer code that can solve directly the full Navier Stokes equations. However, given the state of the art in both algorithms and computer hardware, such Navier Stokes solvers are restricted only to supercomputers, and even then the computation time is quite long.

In order to enable fast and efficient computations, the viscous inviscid interaction code was developed by Cebeci [Ref. 1]. The approach used in this code is to solve the outer flow field using potential methods, and solving the boundary layer flow using a boundary layer method subject to an interaction law, that couples the inner and the outer flows. This interaction law is needed because classical boundary layer methods fail in areas of flow reversal and separation, which are very common in real life flows.

The viscous inviscid interaction code was originally developed, and successfully used, for flows about single airfoils. It was later adapted to cascade flows.

In this thesis the applicability of the code to cascades was investigated by comparing its results to experimental data. It was found that although the code can reasonably predict experimental results in some cases, it still needs improvements before it can be applied generally as a reliable design tool.

A major restriction in improving the code is the lack of a wide data base of appropriate experimental results. Some of the key elements in the code, like transition and turbulence modelling, are based on empirical correlations, and more detailed and accurate experiments should be performed, before a better understanding of these phenomena can be achieved.

In the following, the theoretical background of the code is presented in Chapter II, a description of the code in Chapter III, the results and discussions are presented in Chapter IV and the conclusions and recommendations in Chapter V. A listing of the computer program is given in Appendix A.

II. CASCADE FLOW PROBLEM FORMULATION

This chapter outlines the theoretical background of the viscous inviscid interactive method used in the computer code to investigate cascade flows. Only the major steps in the mathematical developments will be described here. A detailed description of the theory and the numerical methods is given by Cebeci and Bradshaw [Ref. 2] and by Krainer [Ref. 3] on which this chapter is based.

A. INVISCID FLOW METHOD

Inviscid flow is the first building block of the flow and is solved using the panel method. The incompressible two dimensional outer flow must satisfy the Laplace equation:

$$\nabla^2 \Phi = 0 ,$$

subject to the boundary conditions on the surface of the blade:

$$\frac{\partial \Phi}{\partial n} = v_w ,$$

where the commonly used boundary condition of zero normal velocity on the surface is replaced by a specified blowing velocity v_w to allow for the effect of the boundary layer on the outer flow.

In addition, the Kutta condition must be satisfied, in order to prevent the existence of discontinuous pressure distribution near the trailing edge of the blade.

Since the Laplace equation is linear, a solution to a complex flow field can be built by superposition of solutions of elementary flows. The elementary flows used in the panel method are the uniform parallel flow and flows about singularities (sources and vortices).

The panel method is based on replacing each blade by a distribution of sources and vortices on its surface. The surface is divided into a finite number of straight segments, called panels.

On each panel, a uniform source distribution and a uniform vorticity distribution is assumed. The source strength at each panel is set to satisfy the boundary condition at the midpoint of the panel (called the control point), and so, in general the source

strength will vary from panel to panel. The vorticity strength is assumed to be the same for all the panels and is set to satisfy the Kutta condition.

The cascade is defined as an infinite row of similar blades, each one modelled by panels of source and vortex distributions. The flow at each point is found by summing the contributions of all the singularities on all the blades , and the uniform parallel flow.

A useful concept in dealing with such flows is the concept of influence coefficients. An influence coefficient is defined as the velocity at a point induced by a unit strength singularity placed at some other field point. The influence coefficients are a function of geometry and so can be computed for a given cascade and a given choice of panel geometry.

Using the influence coefficients, the normal and tangential velocities at each control point can be written as a function of the unknown source strength of each panel and the unknown vortex strength. Using the boundary conditions, by equating the normal velocity at each control point to the prescribed blowing velocity v_n, and using the Kutta condition (which requires equal velocities on the upper and on the lower panels at the trailing edge), a system of linear equations is constructed.

By solving the system of equations, the strength of the sources and vortices is found, and the velocities (and the pressures) can be computed everywhere in the flow field.

The velocity distribution on the surface of the blade, computed by the panel method, is used as the boundary condition for the boundary layer flow calculations.

It should be noted that in the panel method as used in the present computer code, there is no modelling of the wake, and its effect on the flow field is ignored.

B. VISCOUS FLOW METHOD

Viscous flow is the second building block of the cascade flow and it is applied to the thin boundary layer near the blade surface.

1. Boundary Layer Theory

The boundary layer concept, first suggested by L. Prandtl, assumes that the flow field can be divided into an outer flow where the viscous effects are negligible compared to inertia effects, and a thin layer close to the surface where the viscous effects cannot be neglected. The complete presentation of the boundary layer theory, and the development of the boundary layer equations, is given by Schlichting [Ref. 4].

Under the assumptions of two dimensional incompressible thin boundary layer flow, the Navier Stokes equations and the continuity equation reduce to:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 ,$$

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = u_e \frac{du_e}{dx} + v \frac{\partial}{\partial y} \left(b \frac{\partial u}{\partial y} \right) ,$$

with the boundary conditions:

$$y = 0 \qquad u(x,0) = 0 , v(x,0) = 0 ,$$

$$y = y_e \qquad u(x,y_e) = u_e(x) ,$$

where b denotes $1 + \frac{\nu_t}{\nu}$.

Writing the velocity components in terms of a stream function Ψ :

$$u = \frac{\partial \psi}{\partial y} ,$$

$$v = - \frac{\partial \psi}{\partial x} .$$

This eliminates the continuity equation (which the stream function satisfies by definition).

Introducing the Falkner Skan transformation:

$$\eta = \sqrt{\frac{u_e}{\nu x}}\, y ,$$

$$f(x,y) = \frac{1}{\sqrt{u_e \nu x}} \psi(x,y) ,$$

the momentum equation and the boundary conditions transform to:

$$(bf'')' + \frac{m+1}{2} ff'' + m[1 - (f')^2] = x\left(f' \frac{\partial f'}{\partial x} - f'' \frac{\partial f}{\partial x} \right) ,$$

$$\eta = 0 \qquad f'(x,0) = 0 , f(x,y) = 0 ,$$

$$\eta = \eta_e \qquad f'(x, \eta_e) = 1 ,$$

where m is defined by:

$$m = \frac{x}{u_e} \frac{du_e}{dx} .$$

The third order differential equation can be reduced to a system of first order differential equations by introduction of two new variables U and V :

$$U = f' ,$$

$$V = U' ,$$

$$(bV)' + \frac{m+1}{2} fV + m(1 - U^2) = x\left(U \frac{\partial U}{\partial x} - V \frac{\partial f}{\partial x} \right) ,$$

with the boundary conditions:

$$\eta = 0 \qquad U(x,0) = 0 , f(x,0) = 0 ,$$

$$\eta = \eta_e \qquad U(x, \eta_e) = 1 .$$

The next step is to use a finite difference approach to solve the equations. The box method is applied using central differencing in both the x and η directions, and satisfying the equations midway between nodes.

Applying the box method results in a system of nonlinear equations in the unknown variables (which are f, U and V in each node along the η direction at the current x station).

In order to solve the nonlinear system the Newton iterative procedure is used, linearizing the equations first about the solution at the adjacent upstream station, and then about the preceding iteration. The linearization is performed by letting:

$$f_j^{i,\kappa} = f_j^{i,\kappa-1} + \delta f_j^{i,\kappa} ,$$

$$U_j^{i,\kappa} = U_j^{i,\kappa-1} + \delta U_j^{i,\kappa} ,$$

$$V_j^{i,\kappa} = V_j^{i,\kappa-1} + \delta V_j^{i,\kappa} ,$$

where:

- i denotes location in the x direction
- j denotes location in the y (η) direction
- κ indicates the iteration counter

This linearization results in a system of linear equations for the unknown increments: $\delta f_j^{i,\kappa}$, $\delta U_j^{i,\kappa}$ and $\delta V_j^{i,\kappa}$.

This system of equations is solved repeatedly until the changes in the unknowns are small enough. Since the system is block tridiagonal, Keller's block elimination method is used.

The method described so far, is a direct boundary layer method. It can be used as long as the flow does not separate. Whenever separation or flow reversal occurs, and a zero skin friction coefficient is encountered, the equations become singular and the calculations will break down.

2. Interactive Boundary Layer Method

The interactive boundary layer method is designed to overcome the difficulties encountered at regions of flow reversal and separations. In such areas the external velocity is substantially changed by the viscous effects and can no longer be considered as a known boundary condition for the boundary layer flow.

The general approach to the solution is the same as for the direct method but, since the outer flow is unknown, the velocity at the edge of the boundary layer is written as:

$$u(x, y_e) = u_{e1} + \frac{1}{\pi} \int \frac{d}{d\xi} (u_e \delta^*) \frac{d\xi}{x - \xi} ,$$

where:

1. $u_e(x, y_e)$ is the total velocity at the edge of the boundary layer.
2. $u_{e1}(x)$ is the velocity as computed by the inviscid method.
3. $\frac{1}{\pi} \int \frac{d}{d\xi} (u_e \delta^*) \frac{d\xi}{x - \xi}$ is the Hilbert integral.

The numerical solution of the boundary layer equations follows the same steps as for the direct method, but with some changes.

The transformation of the stream function and the y coordinate uses a constant velocity u_0 as a scaling factor, and a scaled velocity w is introduced:

$$\eta = \sqrt{\frac{u_0}{vx}}\, y\,,$$

$$f(x,\eta) = \frac{1}{\sqrt{u_0 vx}}\, \psi(x,y)\,,$$

$$\mathrm{w} = \frac{u_e(x,y)}{u_0}\,.$$

Using this transformation, the boundary layer equations become a system of first order differential equations:

$$f' = \mathrm{U}\,,$$

$$\mathrm{U}' = \mathrm{V}\,,$$

$$\mathrm{W}' = 0\,,$$

$$(b\mathrm{V})' + \frac{1}{2} f\mathrm{V} + x\mathrm{W}\frac{\partial \mathrm{W}}{\partial x} = x\left(\mathrm{U}\frac{\partial \mathrm{U}}{\partial x} - \mathrm{V}\frac{\partial f}{\partial x}\right),$$

with the boundary conditions:

$$\eta = 0 \qquad \mathrm{U}(x,0) = 0\,, f(x,0) = 0\,,$$

$$\eta = \eta_e \qquad \mathrm{U}(x,\eta_e) = \mathrm{W}(x,\eta_e)\,,$$

$$\mathrm{w}(x,\eta_e) = \frac{u_{el}(x)}{u_0} + \frac{1}{\pi}\int \frac{d}{d\xi}\left(\sqrt{\frac{v\xi}{u_0}}\,[\mathrm{W}(\xi,\eta_e)\eta_e - f(\xi,\eta_e)]\right)\frac{d\xi}{x-\xi}\,.$$

The finite difference box method is used to solve the equations, in the same way as it was used for the direct case, but with two additions:

1. In areas of flow reversal the term $u\partial u/\partial x$ is omitted to assure stable integration (the FLARE approximation).
2. The edge velocity, W_J (where J denotes the edge station) which involves integration, is approximated by :

$$W_J^i = g_i + c_{ii}(W_J^i \eta_J - f_J^i) \ ,$$

where g_i and c_{ii} are obtained from the numerical approximation to the Hilbert integral (which will be presented in the next section).

By using central diffrencing to approximate the differential equations, a system of nonlinear algebraic equations is obtained for the unknown variables (which are f_j, U_j, V_j and W_j). To solve the system of equations, the system is linearized by the Newton iterative procedure, and the resulting linear system is solved (for the new unknown variables which are the increments δf_j^{n+1}, δU_j^{n+1}, δV_j^{n+1} and δW_j^{n+1}).

The solution of the system is repeated until the change in the increments is negligible compared to the preceding iteration, and the whole process is performed again at the next downstream station.

3. Interactive Model

The interactive model is used to couple the boundary layer to the external flow. It is needed in areas where strong interaction occurs, and both the boundary layer and the outer flow must be solved simultaneously. The interaction model provides the outer boundary condition to the boundary layer calculations by adding a correction term to the external velocity computed by the inviscid flow method.

The external velocity is assumed to consist of a potential flow term ($u_{e1}(x)$) and a correction term due to viscous effects ($u_{e\delta}(x)$):

$$u_e(x) = u_{e1}(x) + u_{e\delta}(x) \ .$$

The viscous effect is obtained by a surface distribution of sources on the blade (a concept first suggested by Lighthill [Ref. 5]). The normal velocities at the surface of the blade, induced by these sources, displace the streamlines from the surface in the same way that the actual boundary layer displaces them:

$$\frac{d\delta^*(x)}{dx} = \frac{v(x, \delta^*)}{u_e(x)} \ ,$$

Where $v(x, \delta^*)$ is the normal velocity at the displaced surface.

Assuming that the surface can be approximated to be a flat plate, the normal velocity will be half the local source strength σ (x). Assuming also that the inviscid velocity does not change across the boundary layer, the local source strength will be:

$$\frac{\sigma(x)}{2} = v(x,0) = v(x,\delta^*) - \int_0^{\delta^*} \frac{\partial v}{\partial y}\, dy = \frac{d}{dx}(u_e\delta^*) .$$

The local horizontal velocity induced by the source distribution, is the correction term to the inviscid velocity, and can be represented by the Hilbert integral:

$$\frac{1}{\pi}\int_{x_a}^{x_b} \frac{\sigma(\xi)}{x-\xi}\, d\xi = \frac{1}{\pi}\int_{x_a}^{x_b} \frac{d}{d\xi}(u_e\delta^*)\frac{d\xi}{x-\xi} .$$

The integration is carried out on all the sources on the surface, since the horizontal velocity is influenced by all the sources.

The Hilbert integral is then approximated by a finite series:

$$\frac{1}{\pi}\int_{x_a}^{x_b} \frac{d}{d\xi}(u_e\delta^*)\frac{d\xi}{x-\xi} = \sum_{k=1}^{K} c_{ik}(u_e\delta^*)^k .$$

Where c_{ik} is a matrix of interaction coefficients which are functions of the geometry only (i denotes the chordwise position where $u_{e\delta}$ is evaluated and k is the location of the source which effects $u_{e\delta}$).

Since the computation of $u_{e\delta}$ involves values of δ^* downstream of the current x location, which are not known yet, these terms are taken from the previous iteration using a relaxation formula.

4. Turbulence Model

The turbulence model used here is the algebraic eddy viscosity formulation of Cebeci and Smith [Ref. 6]. According to the model used in the present computer code, the eddy viscosity ν_t is defined by two different expressions, for the inner region and for the outer region:

$$\nu_t = \left\{0.4y\left[1 - \exp\left(-\frac{y}{A}\right)\right]\right\}^2 \left|\frac{\partial u}{\partial y}\right| \gamma_{tr} \quad \text{for} \quad 0 \le y \le y_c ,$$

$$\nu_t = \alpha \int_0^\infty (u_e - u)dy\, \gamma_{tr}\, \gamma \qquad \text{for} \quad y_c \le y \le \delta .$$

Where:

$$A = \frac{26\nu}{\left(\nu \frac{\partial u}{\partial y}\right)^{1/2}} ,$$

$$\gamma = \frac{1}{1 + 5.5(y/\delta)^6} ,$$

$$\alpha = \frac{0.0168}{1 - \beta\left[\frac{\partial u'\partial x}{\partial u'\partial y}\right]^{2.5}} ,$$

$$\beta = \frac{6}{1 + 2R_T(2 - R_T)} \qquad \text{for} \quad R_T < 1 ,$$

$$\beta = \frac{1 + R_T}{R_T} \qquad \text{for} \quad R_T \ge 1 ,$$

$$R_T = \frac{\tau_w}{(-\overline{u'v'})_{max}} .$$

The distance from the wall to the point between the two regions, y_c, is chosen such that the viscosity will be continuous.

The intermittency factor, γ_{tr} is defined by:

$$\gamma_{tr} = 1 - \exp\left[-\frac{u_e^3}{G_\gamma v^2} R_{xtr}^{-1.34} (x - x_{tr}) \int_{x_{tr}}^{x} \frac{d\xi}{u_e} \right].$$

Where:

- R_{xtr} is the Reynolds number based on external velocity and transition location.
- G_γ is an empirical constant, originally assigned the value 1200.

Cebeci and Bradshaw [Ref. 2, p.246] described a different expression for the variable A in the inner region viscosity formula:

$$A = \frac{26v}{(1 - 11.8p^+)^{1/2} \left(v \frac{\partial u}{\partial y} \right)_{max}^{1/2}}.$$

Where:

$$p^+ = \frac{v u_e}{\left(v \frac{\partial u}{\partial y} \right)^{3/2}} \frac{du_e}{dx}.$$

This version of the turbulence model was not implemented in the original computer code. During the work on this thesis, the effect of the modified turbulence model was investigated.

A different intermittency distribution was implemented successfully by Rodi and Schonung [Ref. 7] for transition over separation bubbles. They used for G_γ the expression:

$$G_\gamma = \frac{100}{\exp(0.99 Tu)}.$$

Where Tu is the turbulence level in the free flow. This intermittency model was also investigated during the work on this thesis.

5. Transition

The prediction of transition from laminar to turbulent flow is very difficult and has to rely on empirical correlations. The relation used here to predict the onset of transition is a combination of Michel's method and the e^9 method, and is given by Cebeci and Bradshaw [Ref. 2 , p. 153]:

$$R_{\theta_{tr}} = 1.174\left(1 + \frac{22400}{R_{e_{xtr}}}\right)R_{e_{xtr}}^{0.46}.$$

Where:

1. $R_{\theta_{tr}}$ is the Reynolds number based on the momentum thickness at the onset of transition.
2. $R_{e_{xtr}}$ is the Reynolds number based on x at the onset of transition.

In the computer code, if a laminar separation is detected before transition occurs, the onset of transition is assumed at the point of laminar separation.

III. DESCRIPTION OF THE COMPUTER CODE

The computer code used here to investigate cascade flows was written by Cebeci, and is based on the numerical formulation that was outlined in the previous chapters. In this chapter the general structure and the major subroutines of the code will be described.

A. GENERAL STRUCTURE OF THE MAIN PROGRAM

The main program reads in the cascade data (blade coordinates, spacing and stagger angle), the flow data (inlet angle and Reynolds number), and transition parameters. The transition onset on each surface of the blade can be computed by the program, or can be input by the user. The intermittency parameter G should be specified by the user.

The program then calls subroutine POTNL to compute the outer inviscid flow field for the first cycle. The output of subroutine POTNL is the external velocity distribution on the surface of the blades. This velocity distribution is then transferred to subroutine CASBLP, which calculate the boundary layer flow.

Subroutine CASBLP returns the displacement thickness distribution and the blowing velocity distribution on the blades to the main program. This data is then transferred back to subroutine POTNL to the next cycle of calculations.

The program repeats the cycles of calculations by calling the two subroutines, until the specified number of cycles is reached, or until a convergence criterion is satisfied.

B. DESCRIPTION OF THE SUBROUTINES

1. Subroutine POTNL

This subroutine solves the inviscid outer flow by using the panel method. The subroutine calculates the influence coefficients and calculates the velocities subject to the boundary conditions.

The velocities are evaluated on the displaced surface (the surface created by adding the displacement thickness to the original surface of the blade). The input to this subroutine includes the cascade geometry, the blowing velocity and the displacement thickness (for the first cycle both the displacement thickness and the blowing velocity are taken to be zero).

2. **Subroutine CASBLP**

This subroutine, called by the MAIN program, receives the blade geometry and the velocity distribution as input.

It transforms the x,y blade coordinates to the chordwise tangential coordinates and smooths the velocity data (during the work on this thesis it was found that smoothing the velocity data prevents the detection of the separation bubble near the leading edge, and therefore it was eliminated). The subroutine then calls subroutine COMPBL for further calculations.

3. **Subroutine COMPBL**

This subroutine finds the stagnation point and controls the generation of the boundary layer calculation grid for each surface (the grid starts at the stagnation point and includes 91 points in the chordwise direction for the upper surface and 71 points on the lower surface).

The subroutine then calls subroutine BL2D which calculates the boundary layer parameters for each surface (BL2D is called twice, first for the upper surface and then for the lower surface).

4. **Subroutine BL2D**

This subroutine computes the displacement thickness and the blowing velocity and returns them back to the calling subroutine (COMBL) in arrays compatible with the potential flow calculations (one array that contains all the points of the blade, first the lower surface starting at the trailing edge and proceeding forward, and then the upper surface, starting at the leading edge and proceeding backwards).

BL2D calls the following subroutines:

1. Subroutine INPUT which calculates the following:
   a. NS, the switching point between direct and interactive boundary layer calculations (this point is set at the first pressure peak when the blade is scanned from leading edge towards the trailing edge)
   b. NTR, transition location (only if the transition location is an input. Otherwise it is calculated by subroutine TRNS).
   c. GMTR, the distribution of the intermittency factor γ_{tr}.

   In addition this subroutine generates the boundary layer grid in the η direction and the initial velocity profile, by calling subroutine INTL.
2. CALCIJ, calculates the c_{ij} coefficients used in the Hilbert integral approximation.
3. EDDY, calculates the eddy viscosity (called only after transition has been detected).

4. COEFTR, calculates the coefficients of the boundary layer finite difference equations in transformed form (for the direct method calculations).
5. SOLVE3, solves the linearized boundary layer equations for the F,U and V variables by computing the increments δF, δU and δV

The subroutine then checks the convergence of the Newton iterations and repeats the calculations if needed. If the subroutine detects flow separation or if it reaches the switching point NS, subroutine MAIN2 is called for the interactive method calculations. Otherwise, the subroutine proceeds to the next chordwise point of the grid (NX) and repeats the calculations.

5. Subroutine MAIN2

This subroutine calculates the boundary layer parameters by the interactive method. The subroutine performs the following steps:

1. It first calls subroutines JOINT and COMGI to compute the interaction coefficients.
2. In regions of laminar flow it calls the following subroutines:
   a. COEF, which calculates the coefficients of the boundary layer finite differences equations.
   b. SOLV4, solves for the variables F, U, V and W by computing the increments δF , δU , δV and δW .
   c. TRANS, to check if the condition for transition is satisfied (it also checks for laminar separation and initiates transition at the point of laminar separation if it is detected).

   The subroutine then checks for convergence of Newton iterations and repeats the calculations as needed.
3. In regions of turbulent flows the subroutine calls the following subroutines:
   a. EDDY, to compute the eddy viscosity parameter B ($B = 1 + \nu_t/\nu$).
   b. COEF and SOLV4, the same as for laminar flow.

6. Subroutine OUTPUT

This subroutine computes the boundary layer parameters. It is called with a parameter "INDEX" which determines the type of calculations:

1. For INDEX = 1 the computations relates to transformed coordinates (direct boundary layer method) using the relations:

$$c_f = \frac{2\,V(1,2)\,B(1,2)}{\sqrt{R_{ex}}} ,$$

$$V_w = u_e\sqrt{\frac{u_e}{x}}\,V(1,2)\,,$$

$$D = u_e\delta^*\sqrt{R_e}\,,$$

$$\delta^* = \frac{x}{\sqrt{R_{ex}}}\,(\eta(NP) - F(NP))\,,$$

$$\theta = \frac{x}{\sqrt{R_{ex}}}\sum_{j=2}^{NP} a_j\,U_j\,(1 - U_j)\,.$$

Where V(1,2) and B(1,2) are the velocity gradient and the viscosity parameter at the surface, respectively, and η(NP) and F(NP) are η and F evaluated at the edge of the boundary layer.

2. For INDEX = 2 the subroutine calculates the boundary layer parameters for semi--transformed coordinates (interactive boundary layer method) using the relations:

$$C_f = \frac{2\,V(1,2)\,B(1,2)}{\sqrt{x}\,\sqrt{R_e}\,[W(NP)]^2}\,,$$

$$V_w = \frac{V(1,2)}{\sqrt{x}}\,,$$

$$u_e = U(NP)\,,$$

$$D = (U\eta - F)\sqrt{x}\,,$$

$$\delta^* = \left(\eta - \frac{f}{U}\right)\sqrt{x}\,.$$

For NX > NTR (after transition has been detected), subroutine SMPSON is called (subroutine SMPSON calculates the coefficient of the outer region eddy viscosity). The subroutine then prints out the velocity profiles at the required stations.

7. Subroutine TRANS

This subroutine calculates the transition location based on the Michel criterion or based on laminar separation (whichever occurs first). If transition has been detected the intermittency distribution is calculated for all the remaining points of the surface.

8. Subroutine FILLUP

This subroutine increases the number of points in the boundary layer grid (in the η direction) as needed. It also fills up the arrays of F, U, B, W and V between the edge of the boundary layer to the end of the arrays (with V = 0, W,B and U, with the last values that they had in the edge of the boundary layer and F as the integral of U).

9. Subroutine EDDY

This subroutine calculates the eddy viscosity using the Cebeci--Smith two layer eddy viscosity formula. It receives the vectors U,V and η at a point and computes the viscosity vector B.

10. Subroutine INTL

This subroutine generates the boundary layer grid in the η direction. It sets the number of grid points and generates the initial velocity profile.

IV. RESULTS AND DISCUSSION

The viscous inviscid interaction code was run with several cascades on which experimental data is available. In order to enable a thorough comparison between experimental results and the computed results, a very detailed experimental data base is needed. The data should include measurements of the boundary layer development along the blade, velocity profiles along the boundary layer, transition location and distribution, flow separation, and external velocity distribution.

Unfortunately, very few cascade experiments has been performed, which obtained the required data with sufficient accuracy, due mostly to the lack of appropriate measurement equipment. Only recently, with the introduction of non--interfering methods like the Laser Doppler Velocimeter (LDV), the required data can be measured accurately.

Recently an experiment involving the investigation of a linear compressor cascade of Controlled Diffusion Blading (which will be referred here as the CD cascade) has been carried out by Elazar [Ref. 8]. Most of the work in the present thesis, involves comparison of the computer code results with Elazar's experimental results.

Other cascades that were investigated are:

1. A shockless, supercritical airfoil cascade, designed in 1974 by Korn in cooperation with Pratt & Whitney Aircraft (referred here as the P & W cascade). The experimental results of the cascade were obtained from a report by Hobbs, Wagner, Dannenhoffer and Dring [Ref. 9].
2. Stator blade of a single stage axial compressor (referred here as the C4 cascade). The blade profile is the British C4 section (10% thickness) on a circular arc camber line. The experiment has been performed by Walker [Ref. 10]. The detailed boundary layer measurements are not presented in the report and were obtained directly from the author.

The code failed to run with two other cascades:

1. A highly loaded, double circular arc blade with a sharp leading edge and a sharp trailing edge, used in a compressor cascade that was investigated by Deutsch and Zierk [Ref. 11].
2. V2 double circular arc blade, highly loaded cascade. This cascade was investigated by Hoheisel and Seyb [Ref. 12].

In both cases the code calculated the potential flow successfully but failed in trying to compute the first cycle of the boundary layer calculations.

A. CD CASCADE

The experimental data for the CD cascade was obtained at $M = 0.25$, $R_e = 700000$ and at three inlet angles: 40° (the design condition), 43.4° and 46° . The spacing was 0.6 of the chord and the stagger angle 14.27° . A general layout of the cascade is shown in Figure 1 on page 20.

The following observations were concluded from the experiment:

1. A separation bubble exists near the leading edge on the upper surface at all the inlet angles. The bubble became larger at increased inlet angles.
2. Transition from laminar to turbulent flow occurred above the separation bubble (on the upper surface).
3. Transition on the lower surface occurred at midchord.
4. The boundary layer thickness on the upper surface increased with inlet angle, and reached a thickness of 15% chord at the highest inlet angle. The boundary layer thickness on the lower surface did not change significantly with inlet angle.
5. The turbulent boundary layer on both surfaces remained fully attached at all the inlet angles.

1. Transition location and intermittency distribution

The effects of the transition location and the intermittency factor were investigated. The code was first run with the transition location calculated by the code, and with several values of the intermittency factor G_γ. It was found that the code did not run with $G_\gamma = 1200$ (which is the value used usually for high Reynolds numbers). The highest value of G_γ with which the code run successfully was 900.

The code failed to predict the separation bubble on the upper surface, and predicted laminar separation at 78% chord on the lower surface (which did not occur in the experiment). Transition on the upper surface occurred at 41% chord (detected by Michel's criterion) and at 78% chord on the lower surface (at laminar separation).

The shape factor computed by the code was compared to the experimental results. As can be seen in Figure 2 on page 21 the shape factor as predicted by the code deviates substantially from the actual results, due mainly to the different transition location.

On the lower surface, as can be seen in Figure 3 on page 22 the shape factor deviates even more from the experimental results. In this figure the effect of changing the intermittency factor G_γ can be seen. For both the extreme values of G_γ , 10 and 900, the computed shape factor curve is far from agreement with the actual results.

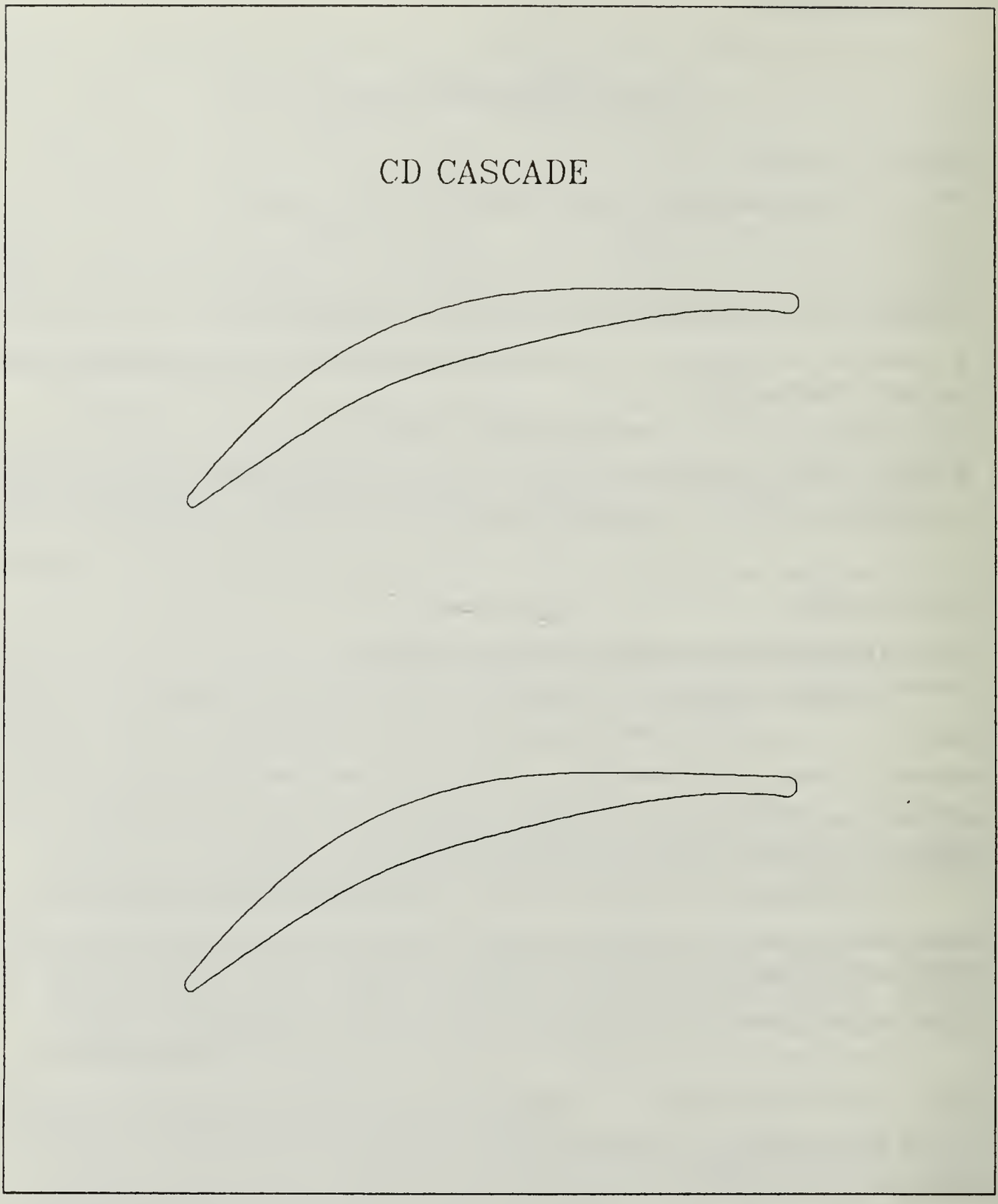

Figure 1. Controlled Diffusion cascade

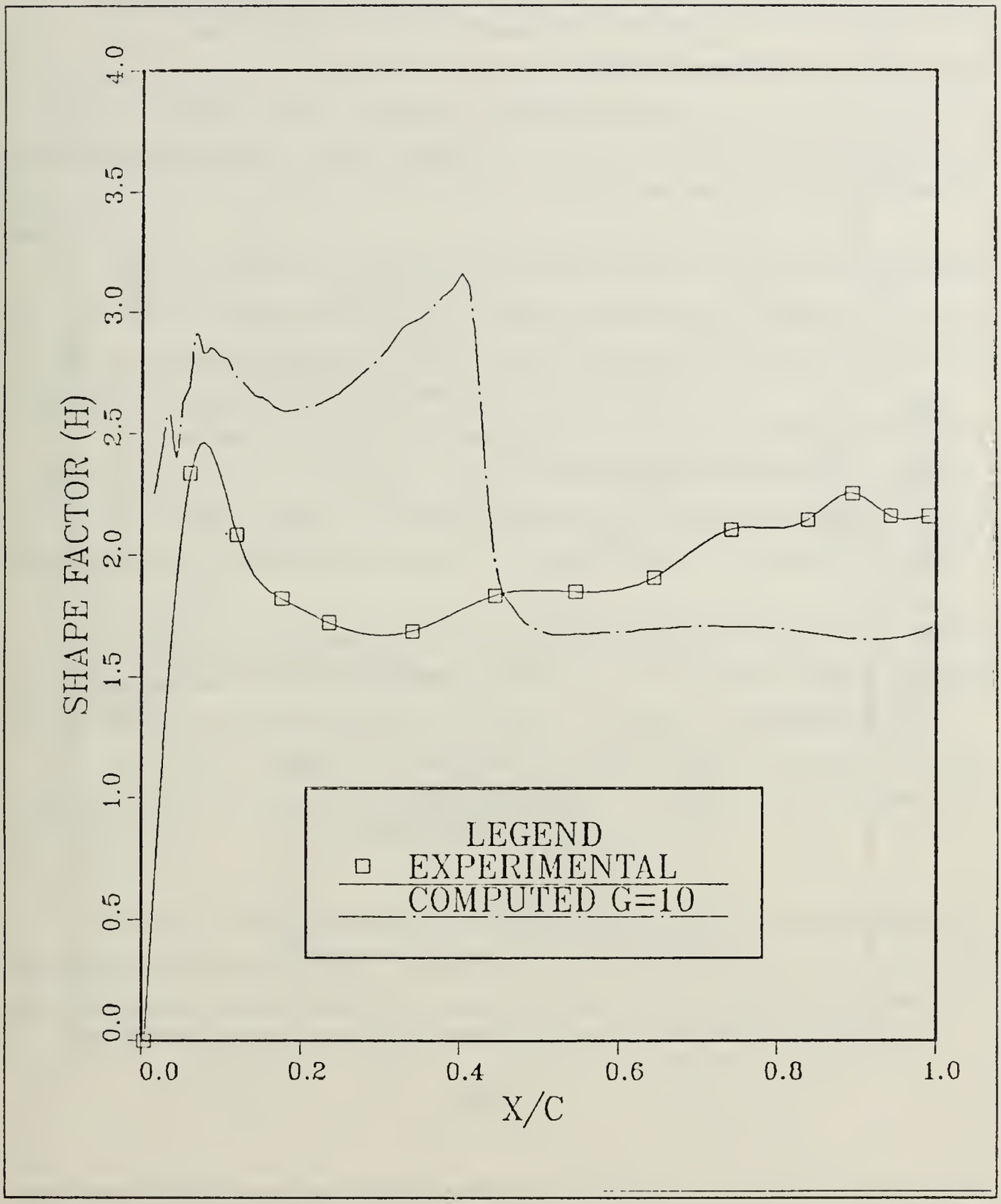

Figure 2. Shape factor comparison on the upper surface: Transition computed by the code ($\beta = 40°$)

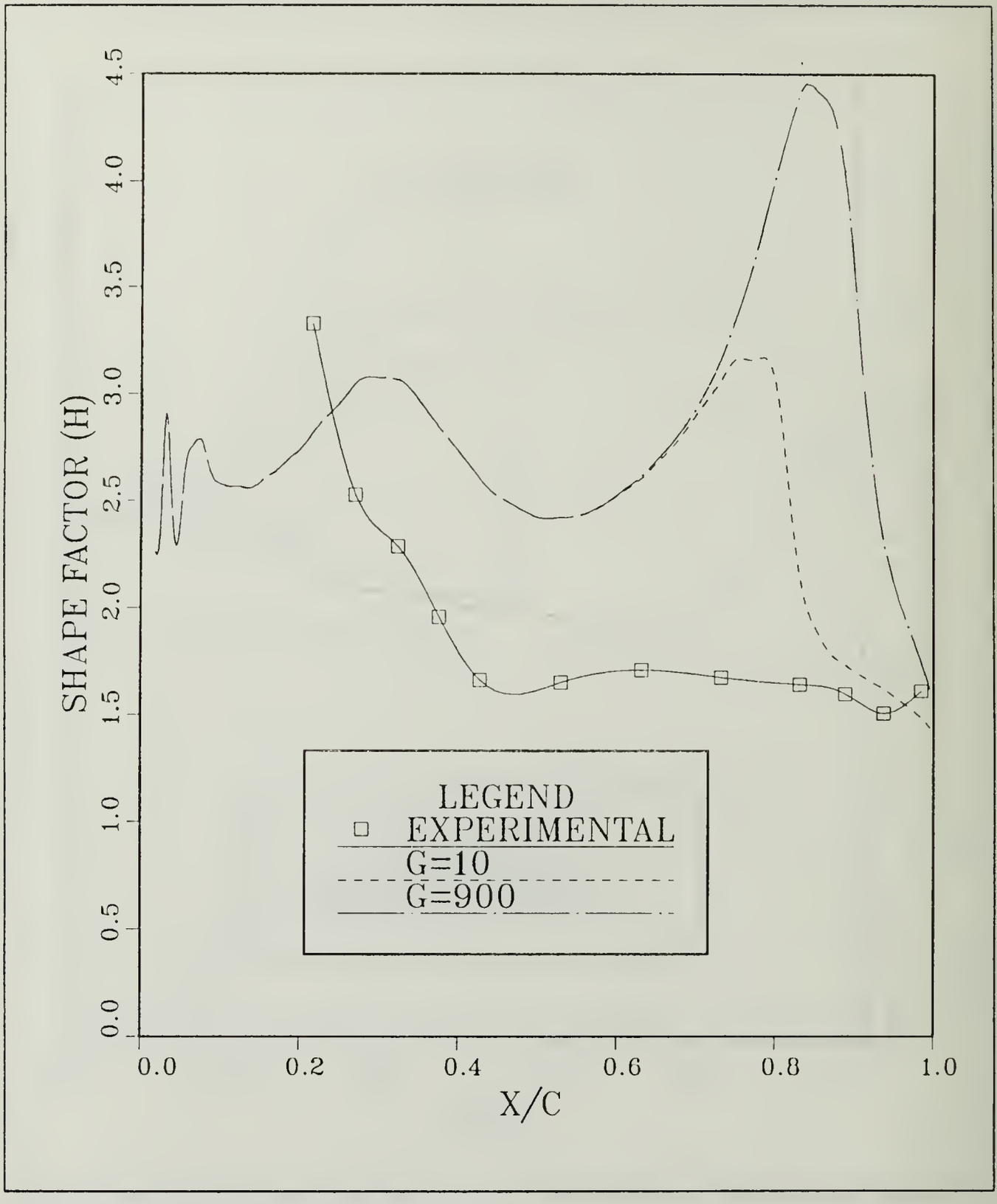

Figure 3. Shape factor comparison on the lower surface: Transition computed by the code ($\beta = 40°$).

Since the code did not predict the existence of the separation bubble on the upper surface, it was decided to try to eliminate the smoothing of the external velocity as computed by the potential flow subroutine (originally, the velocities were smoothed in subroutine CASBLP prior to boundary layer calculations). It was found that without smoothing the velocities a small separation bubble is predicted by the code at 4% of chord. The onset of transition is set by the code at the beginning of the separation bubble.

The code was run with unsmoothed velocities with two values of G_γ, 10 and 900. The shape factor behavior can be seen in Figure 4 on page 24. Changing the value of G_γ did not change the shape of the curve much, and generally the shapes of the computed and the experimental curves look alike.

The elimination of the velocity smoothing in the code, also affects the thickness of the boundary layer. In Figure 5 on page 25 the displacement thickness is plotted for both cases (with and without the velocity smoothed). Without smoothing, the displacement thickness is much thicker, especially on the rear half of the blade, which is closer to the actual results.

The effect of changing the intermittency distribution to the one used by Rodi and Schonung [Ref. 7] was investigated. It was found, as can be seen in Figure 6 on page 26 that the effect of the new model is equivalent to using G_γ in the present model.

On the lower surface it was necessary to run the code with transition as input, to get reasonable results, as can be seen in Figure 7 on page 27 for transition input at 21% of chord.

At the off design conditions (inlet angles of 43.4° and 46°) a similar behavior of the transition has been observed, as can be seen for example in Figure 8 on page 28 for the upper surface and in Figure 9 on page 29 for the lower surface, both at $\beta = 46°$.

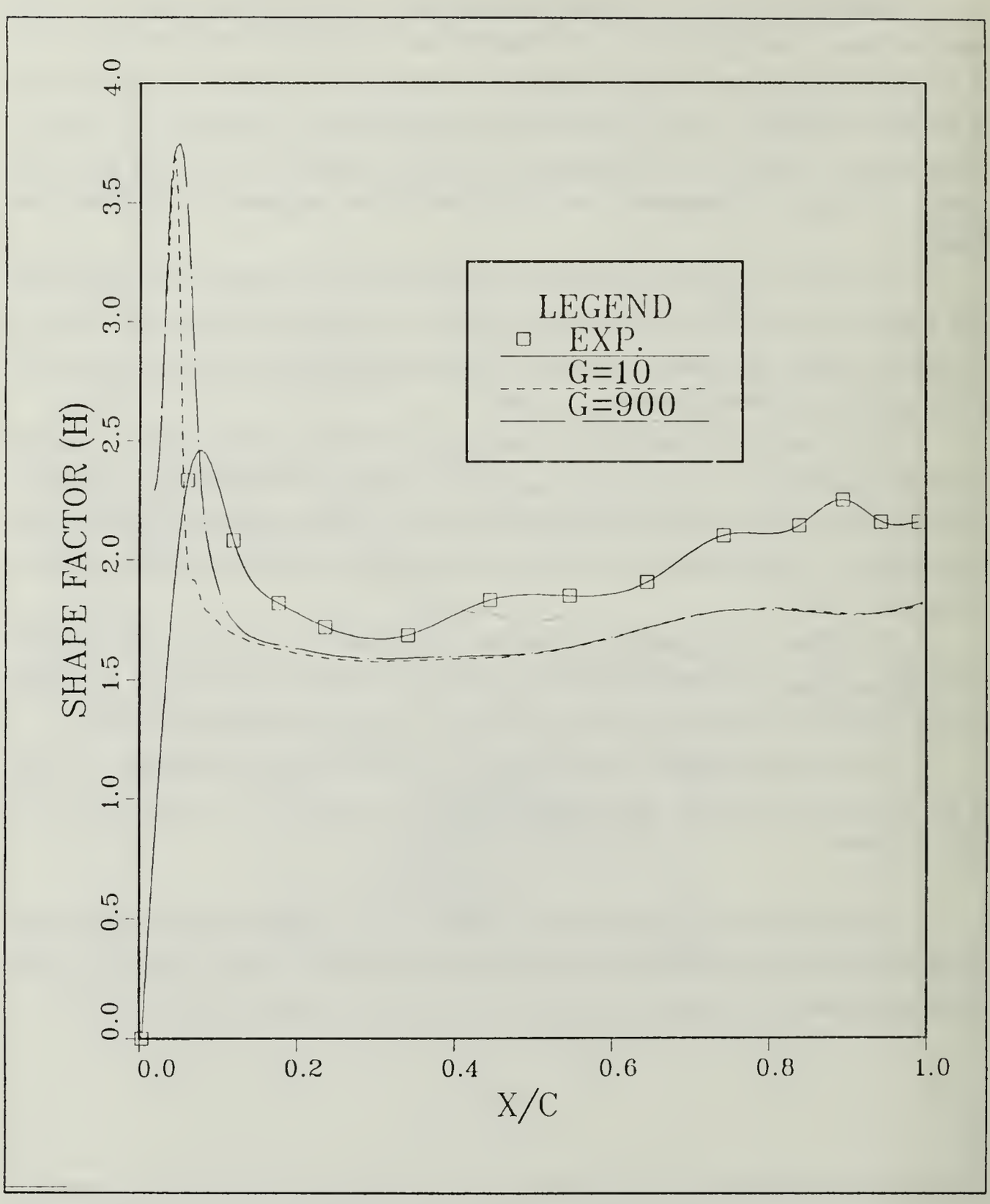

Figure 4. Shape factor on the upper surface without velocity smoothing.

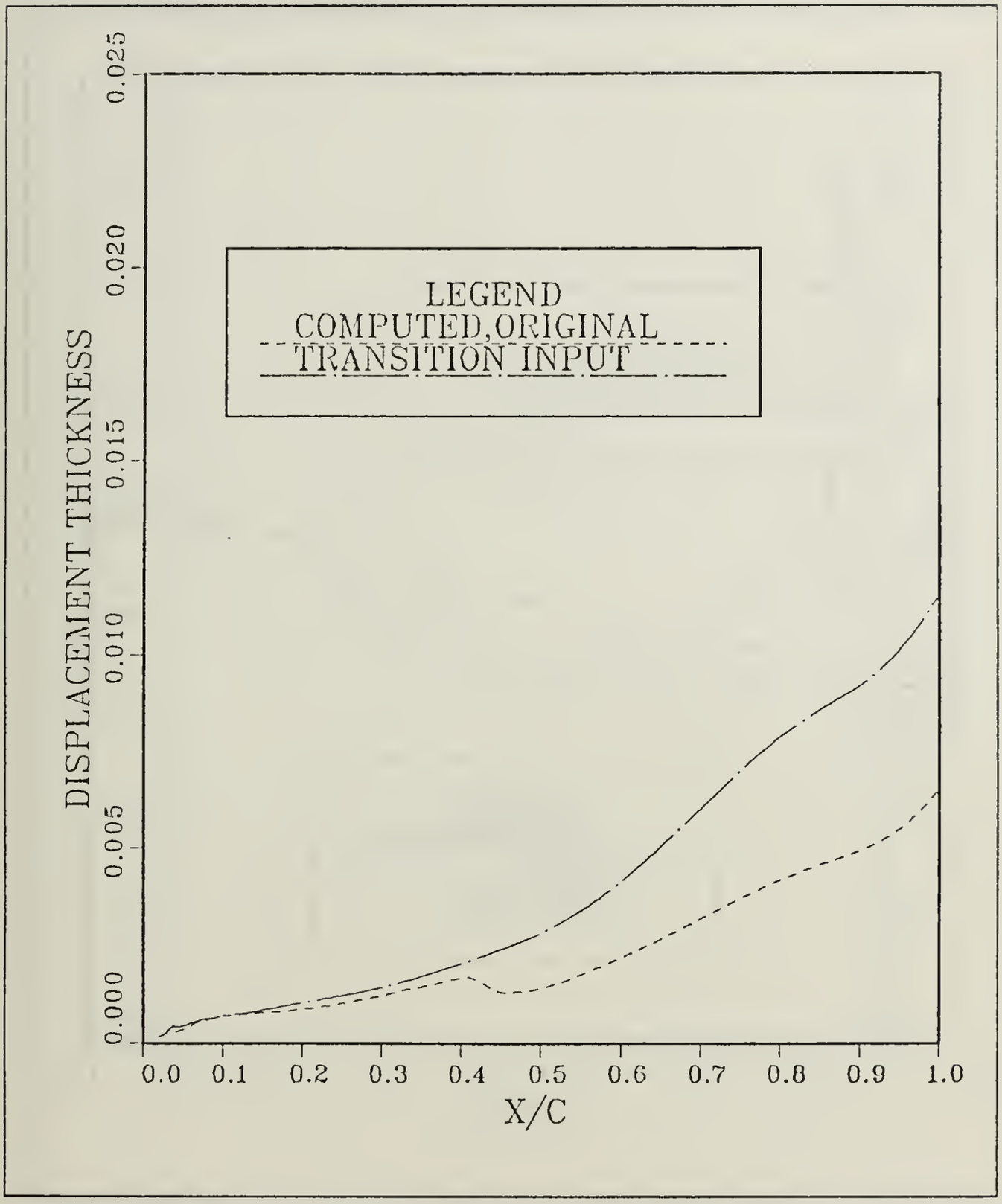

Figure 5. Displacement thickness: The effect of velocity smoothing.

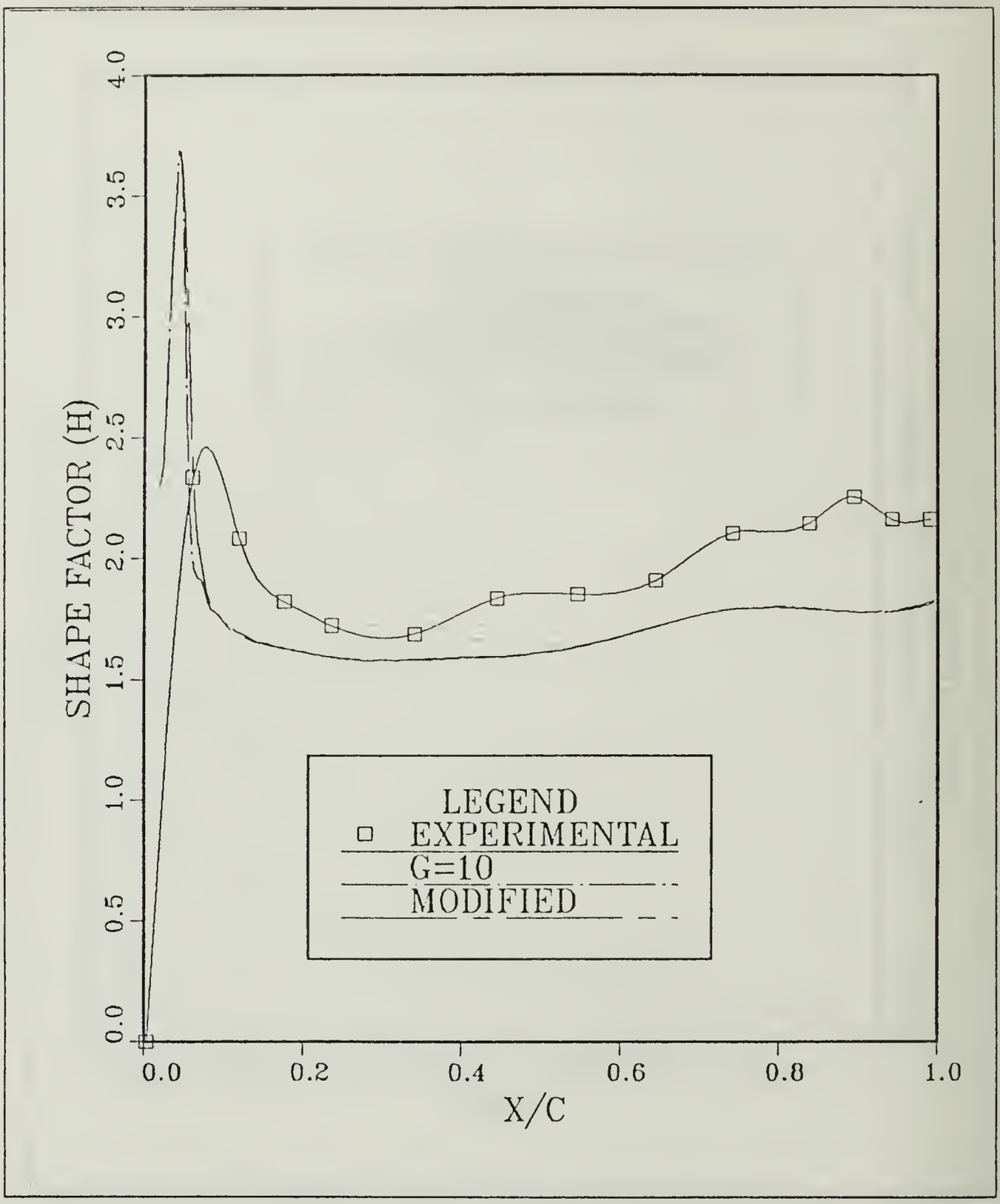

Figure 6. The effect of the intermittency model: Upper surface, $\beta = 40°$

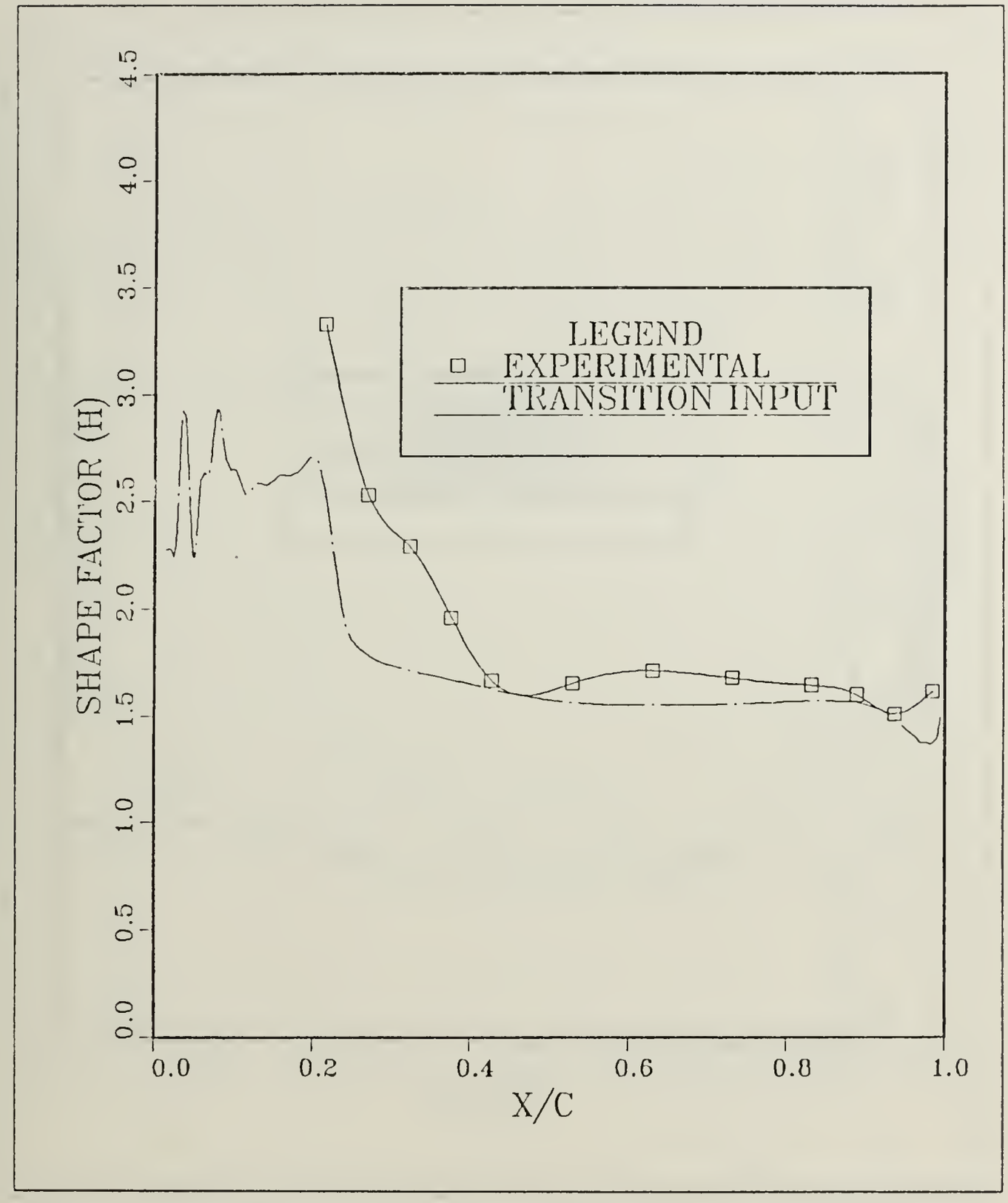

Figure 7. Shape factor on the lower surface with transition input at 21%.

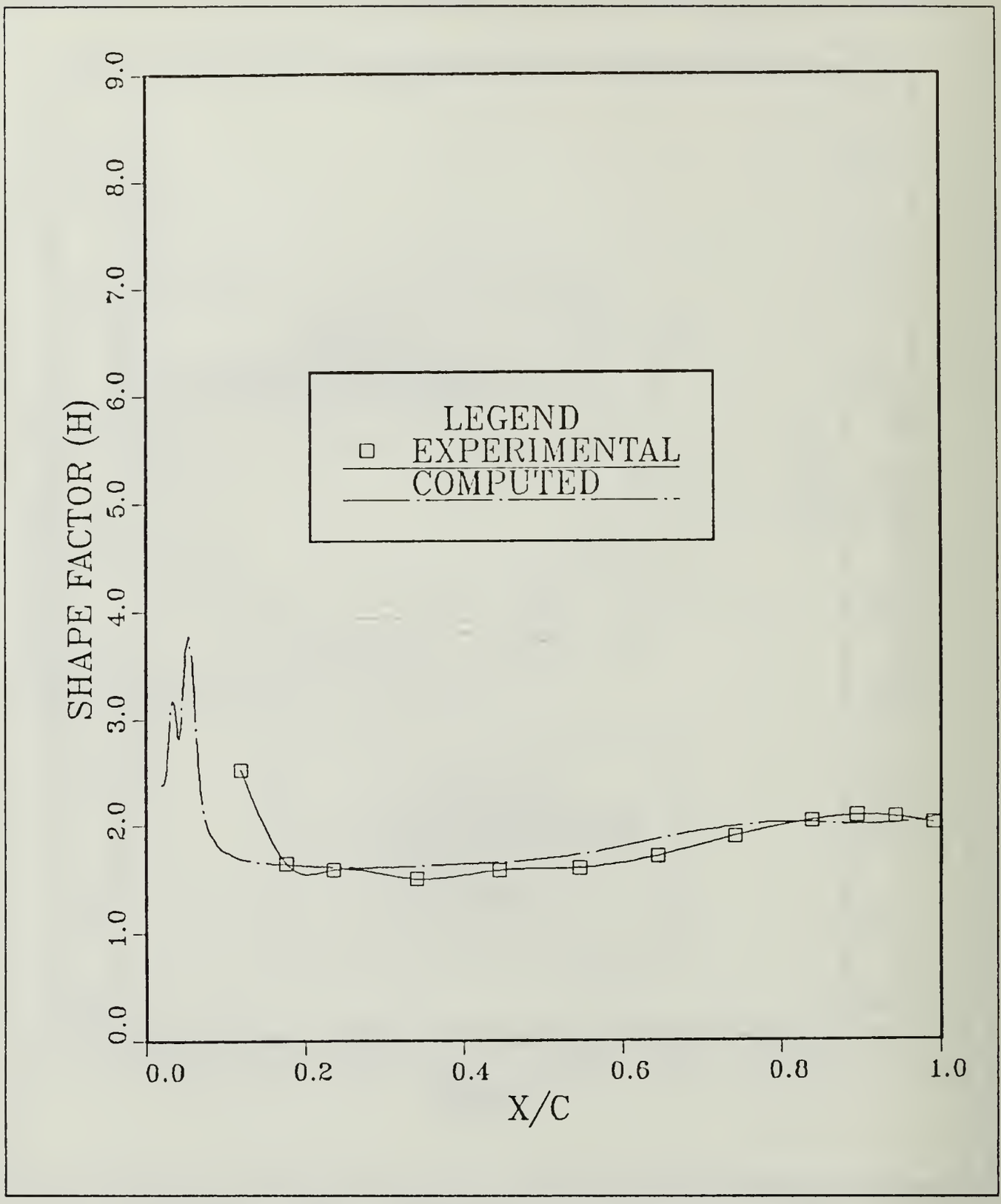

Figure 8. Shape factor at $\beta = 46°$ on the upper surface

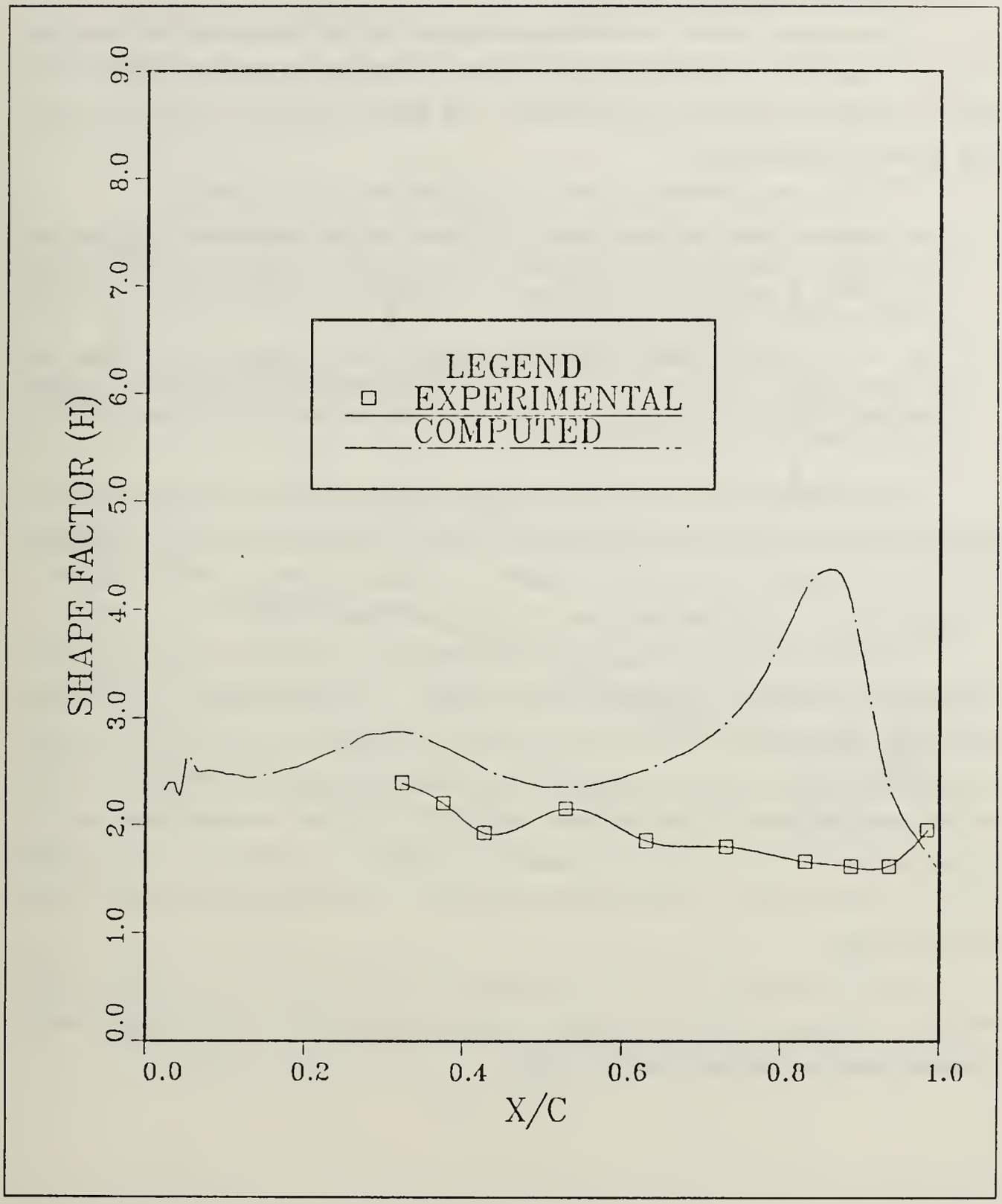

Figure 9. Shape factor at $\beta = 46°$ on the lower surface.

2. External Velocity Distribution

The external velocity distribution, computed by the code using the interaction law, was compared to the experimentally measured velocities. It was found that in general, the velocities measured experimentally, were higher than those computed by the code for all the inlet angles.

There are two possible sources to the discrepancy in the velocities:

1. The computer code calculates pure 2--D flows. In the experiment the flow was observed to accelerate due to the effect of the boundary layer on the side walls (a 3--D effect). This effect was calculated in the experiment and is referred to as the AVDR correction [Ref. 8, p.43].
2. The flow accelerates due to the thickening of the boundary layer. Since the boundary layer as computed by the code is substantially thinner than the actual boundary layer (as will be discussed in the next section) the external velocities predicted by the code are smaller.

To compensate for the first error source, all the computed velocities were compared to the experimental velocities corrected by the AVDR correction. The comparison between the velocities can be seen in Figure 10 on page 31 for $\beta = 40°$, in Figure 11 on page 32 for $\beta = 43.4°$ and in Figure 12 on page 33 for $\beta = 46°$.

It can be seen from the figures that the difference between the computed and the experimental velocities is larger on the lower surface. The reason might be the method with which the correction to the inviscid velocity is computed. The assumption on which the interaction law is based, is that only sources (representing the viscous effects) on the surface being considered, affect the local velocity. In reality, the boundary layer on both surfaces affects the local velocity (because the boundary layer developed on the upper surface of a blade, causes a velocity disturbance that is felt on the lower surface of the adjacent blade).

Since the boundary layer on the lower surface is much thinner, its effect on the velocity on the upper surface is much smaller than the effect of the upper surface boundary layer on the lower surface velocity.

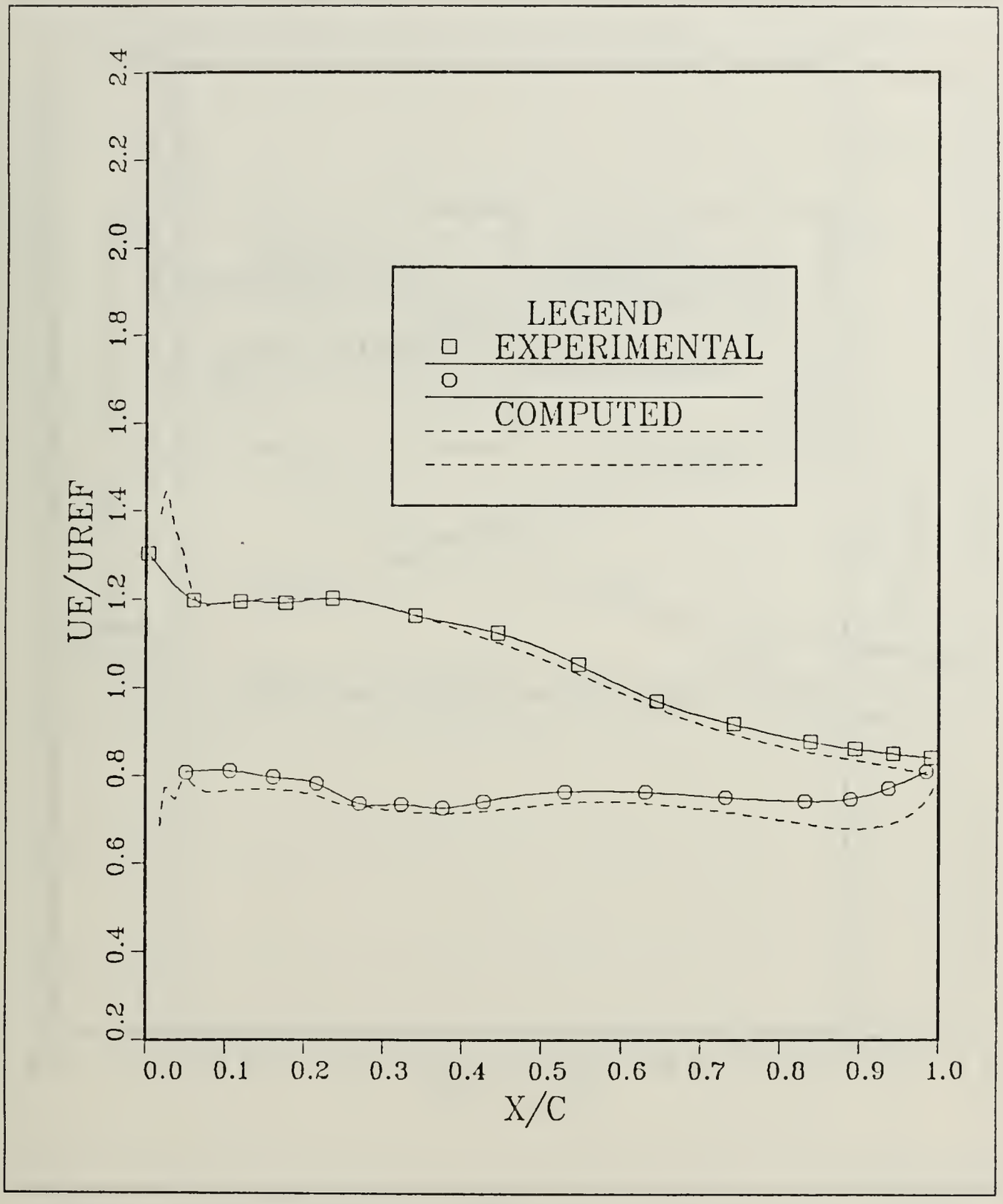

Figure 10. External velocity at $\beta = 40°$

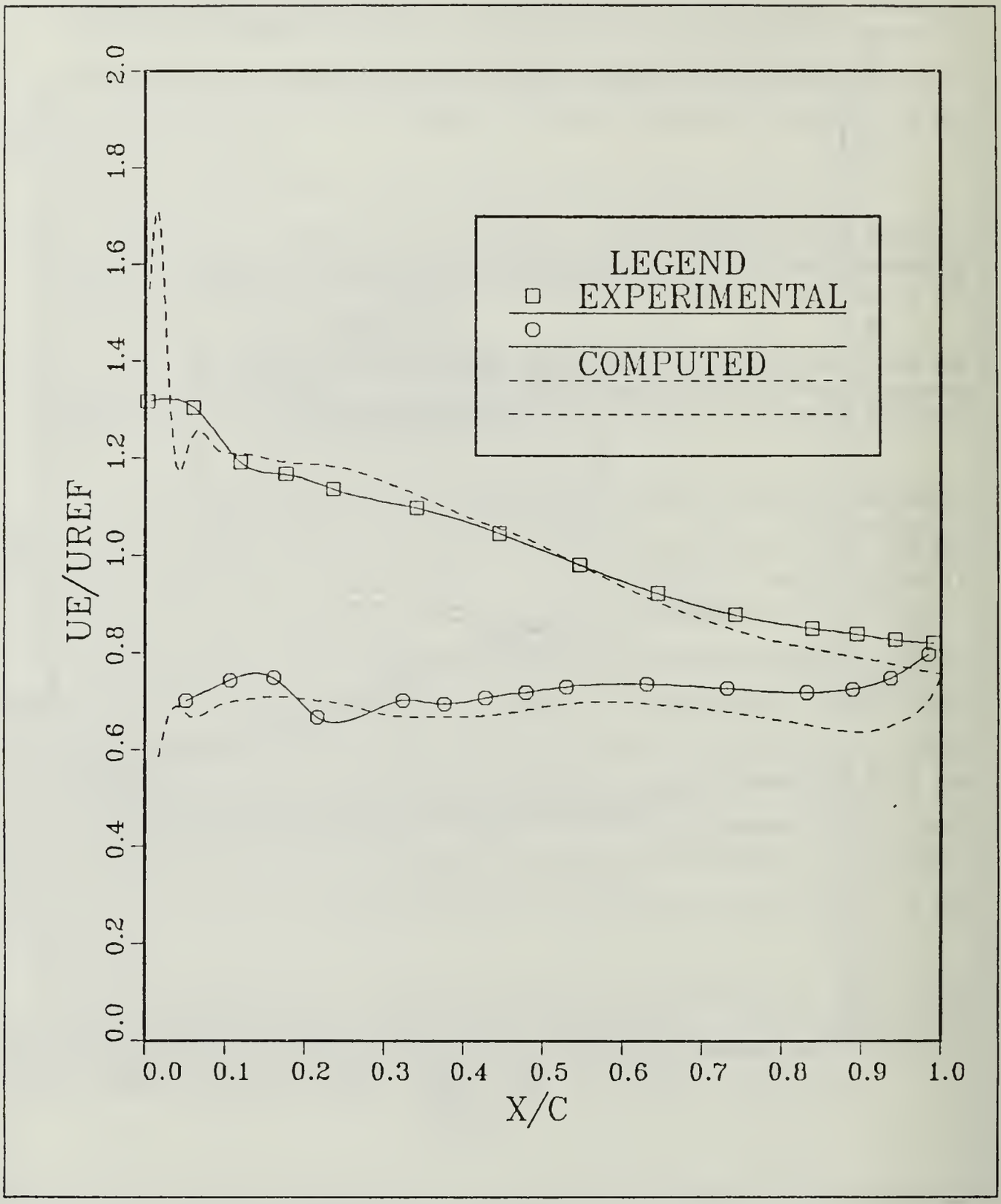

Figure 11. External velocity at $\beta = 43.4°$

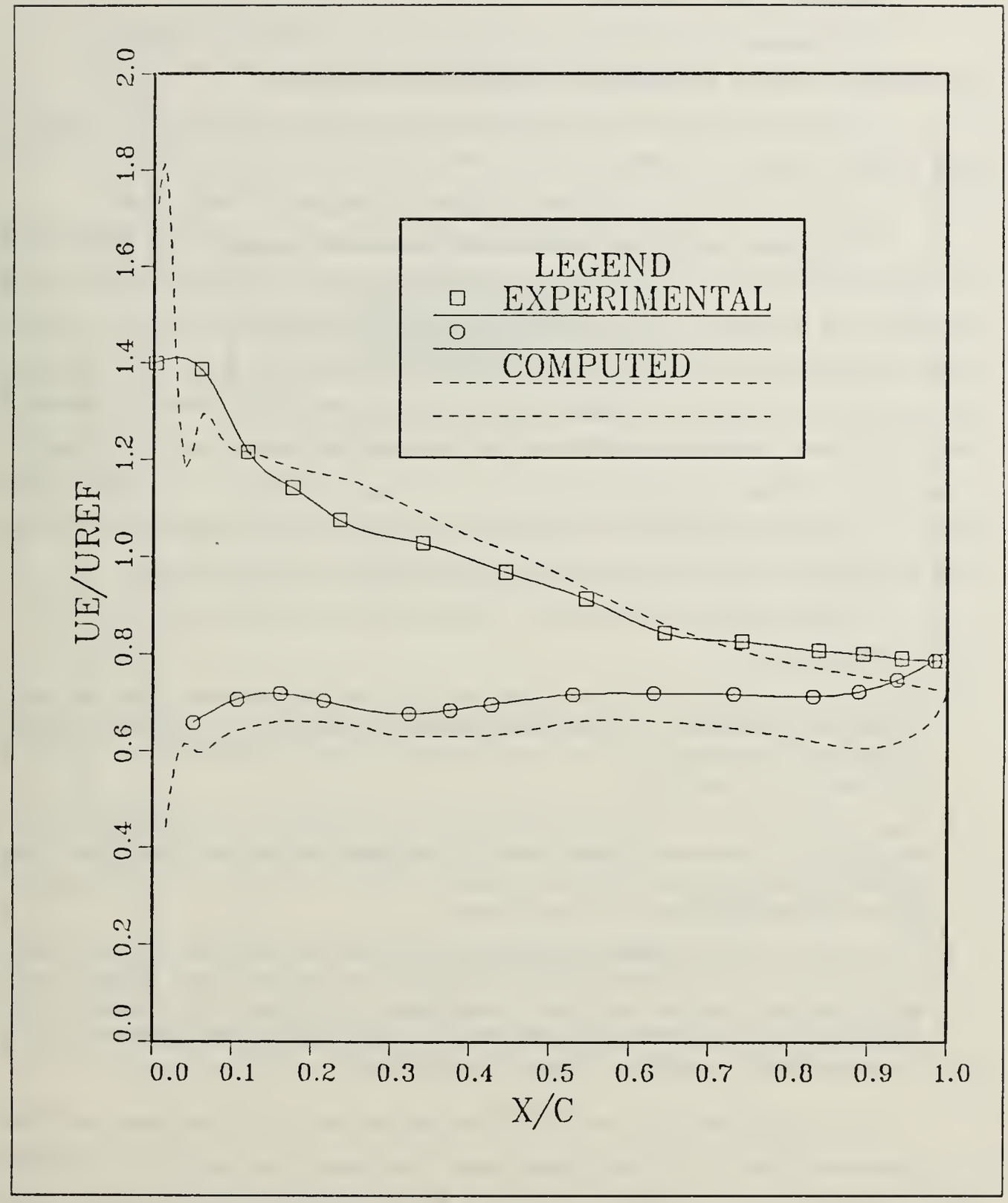

Figure 12. External velocity at $\beta = 46°$

3. Boundary Layer Thickness

The boundary layer thickness as computed by the code was compared with the experimental results by comparing the displacement thicknesses.

It was found that on the lower surface the computed and the actual displacement thickness agree quite well, as can be seen in Figure 13 on page 35 for $\beta = 40°$, Figure 14 on page 36 for $\beta = 43.4°$ and in Figure 15 on page 37 for $\beta = 46°$.

On the upper surface the displacement thicknesses computed by the program are significantly thinner than those measured experimentally. The difference between the computed and the actual thickness increases along the blade and it increases with increased inlet angle. It was found that by using a different expression for the inner region eddy viscosity (as mentioned in chapter II), the displacement thickness can be increased, but the difference between the actual and the computed thickness is still substantial, especially at the higher inlet angles. Figure 16 on page 38, Figure 17 on page 39 and Figure 18 on page 40 shows the displacement thickness on the upper surface for the three inlet angles, with the original and the modified eddy viscosity models.

The large error in the prediction of the boundary layer thickness, can be the result of several reasons:

1. The transition model used in the code, sets the onset of transition at the first point of laminar separation. It causes rapid transition to turbulent flow which reattaches immediately, resulting in a very small separation bubble compared to the bubble observed in the experiment.
2. The turbulent model used in the code could be inaccurate. It was derived based on empirical data obtained in single airfoil experiments and not with cascades. In addition the present model does not include the effects of the free stream turbulence (that was relatively high in the experiment).
3. The boundary layer as measured in the experiment was quite thick, especially at the higher inlet angles (it reached 15% of the chord at $\beta = 46°$). Such a thick boundary layer may violate the basic assumptions on which the boundary layer equations, and the interaction law, were based (especially when the spacing between the blades is small, 60% chord in this case).

It was suggested that one of the possible reasons to the inaccurate prediction of the boundary layer is the blunt trailing edge of the blade, that might cause difficulties in the computations. A modified blade, with a sharp trailing edge has been run, and the displacement thickness distribution can be seen in Figure 19 on page 41. As can be seen in the figure the sharp trailing edge affects only the boundary layer adjacent to the trailing edge, and therefore cannot provide an explanation to the difference between the actual and the computed displacement thickness.

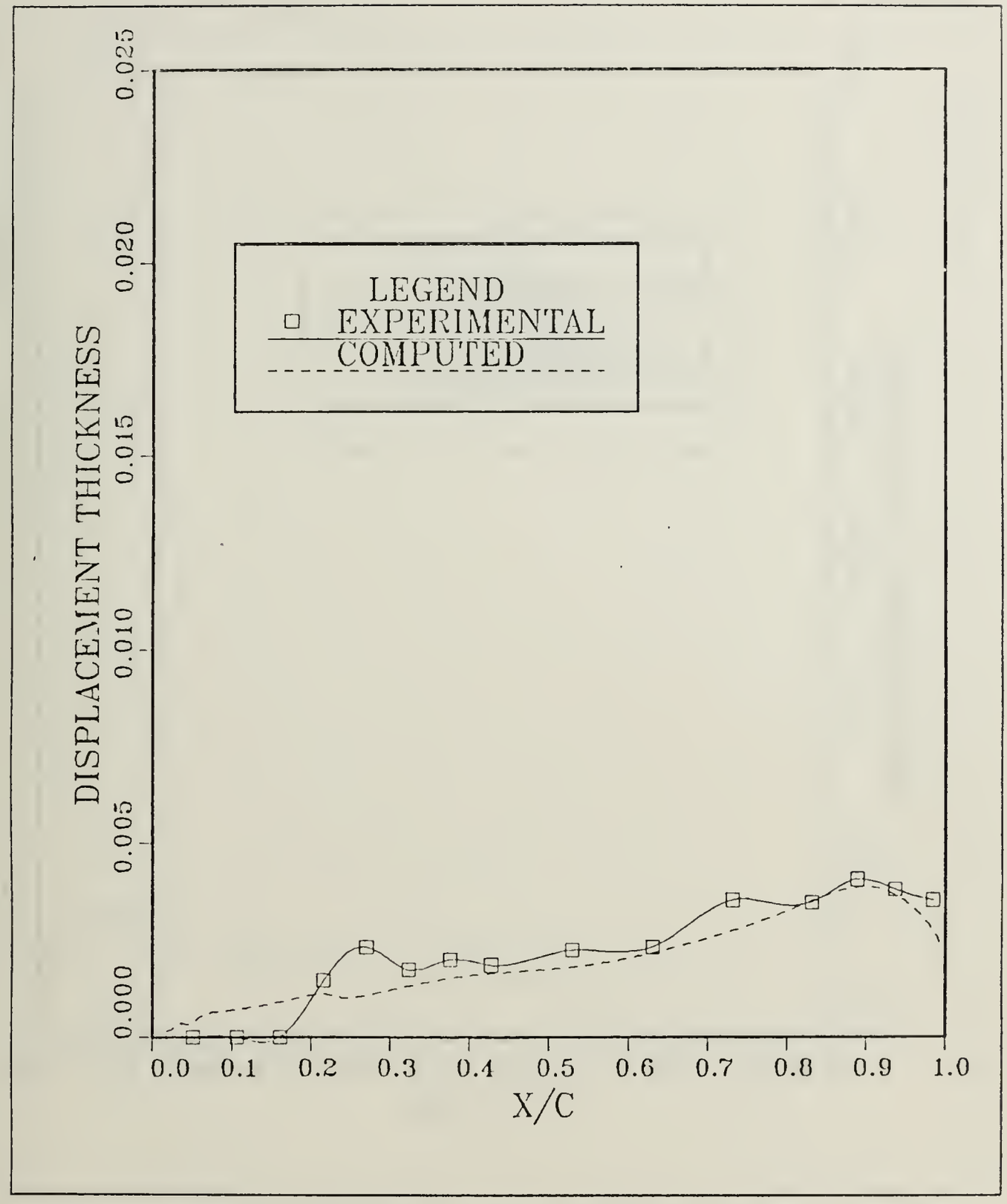

Figure 13. Displacement thickness on the lower surface ($\beta = 40°$)

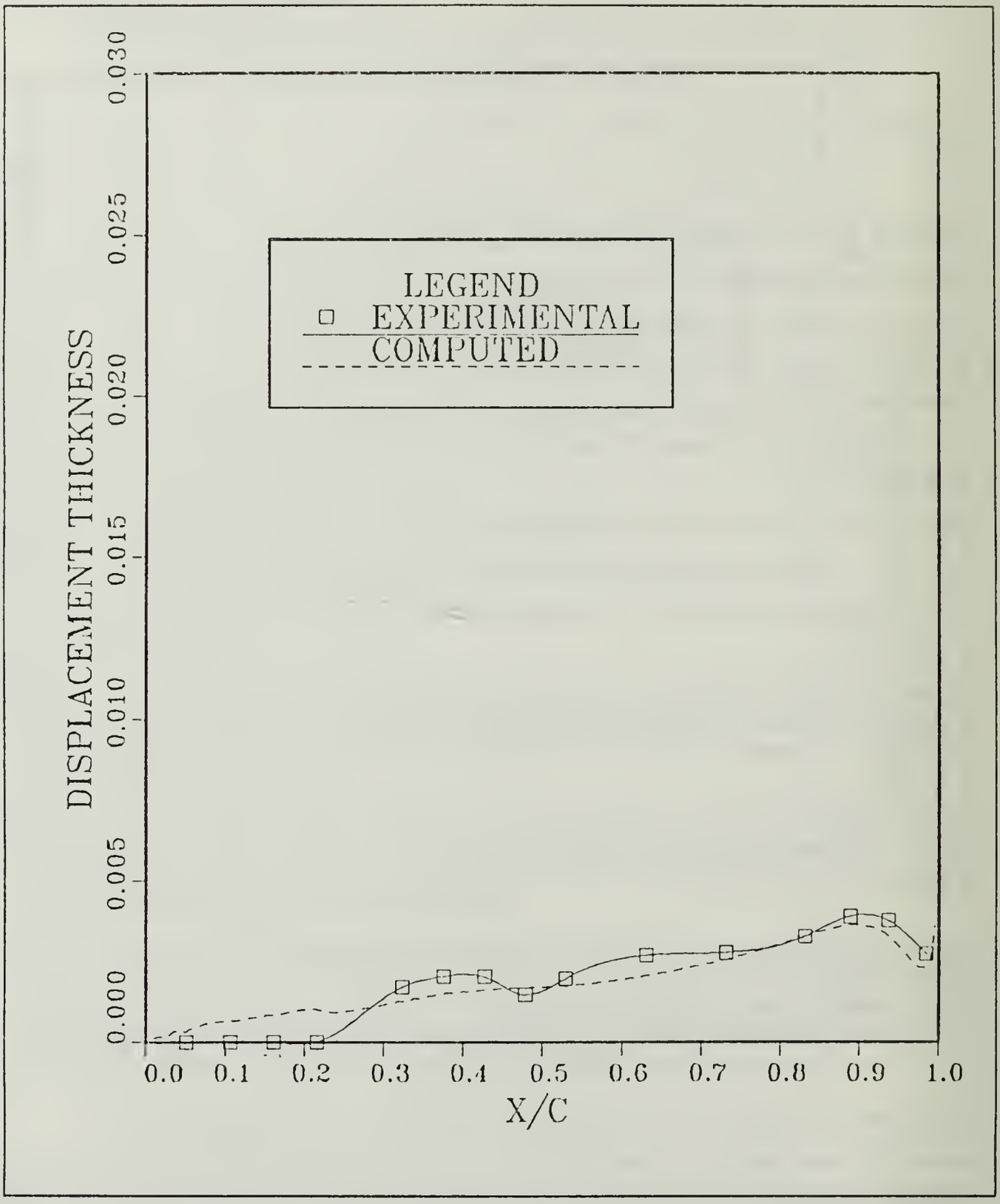

Figure 14. Displacement thickness on the lower surface (β = 43.4°)

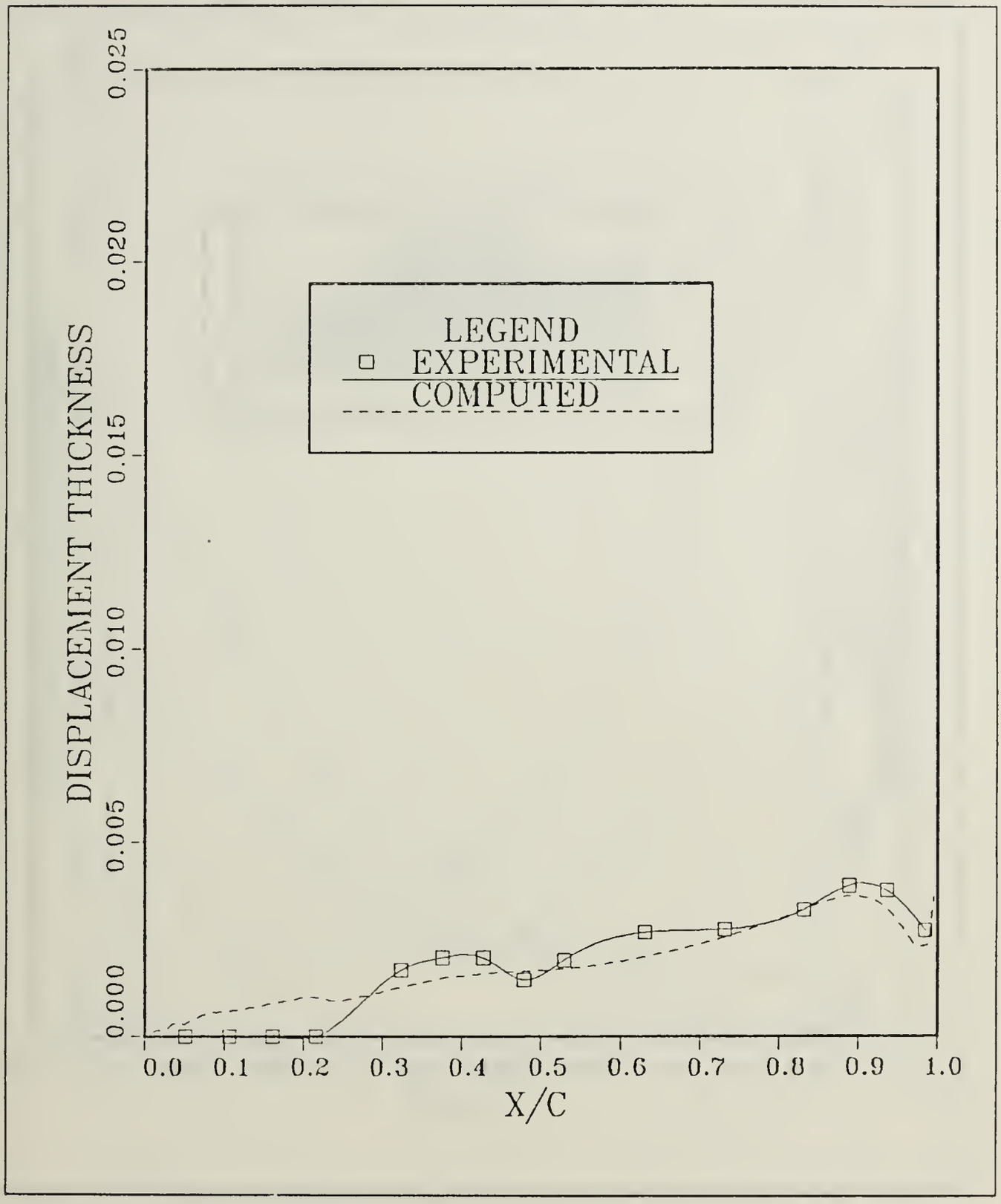

Figure 15. Displacement thickness on the lower surface ($\beta = 46°$)

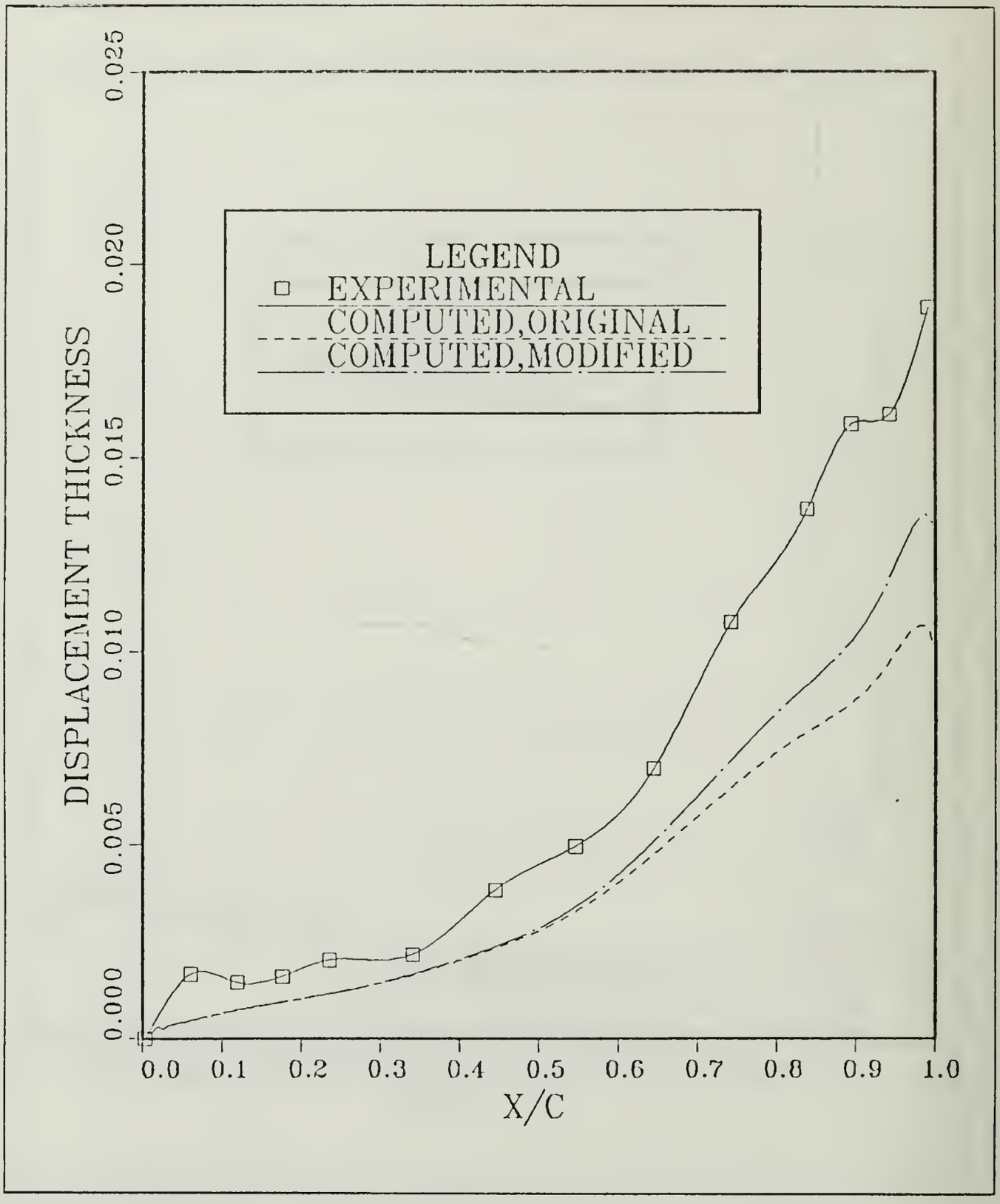

Figure 16. Displacement thickness on the upper surface (β = 40°)

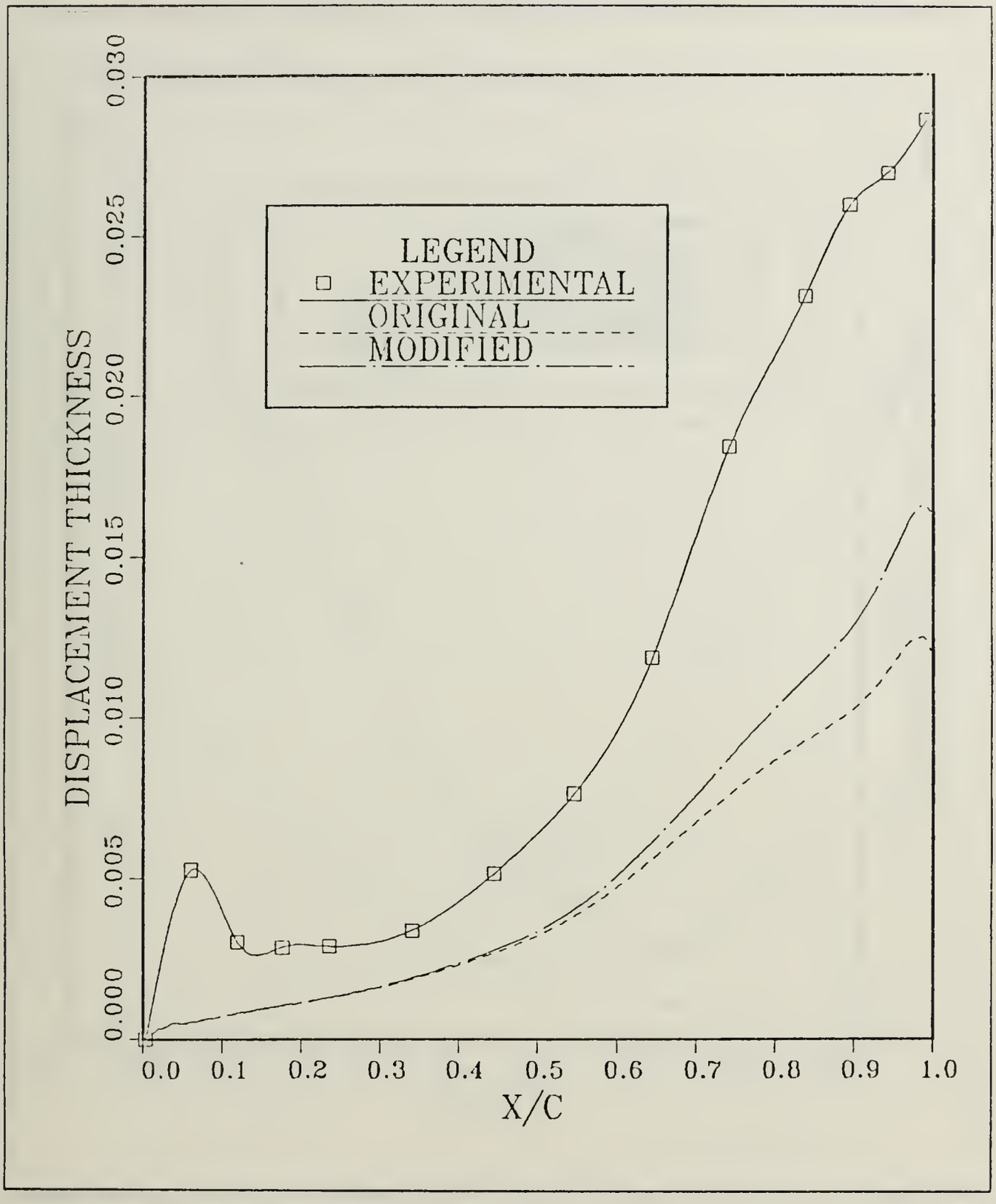

Figure 17. Displacement thickness on the upper surface (β = 43.4°)

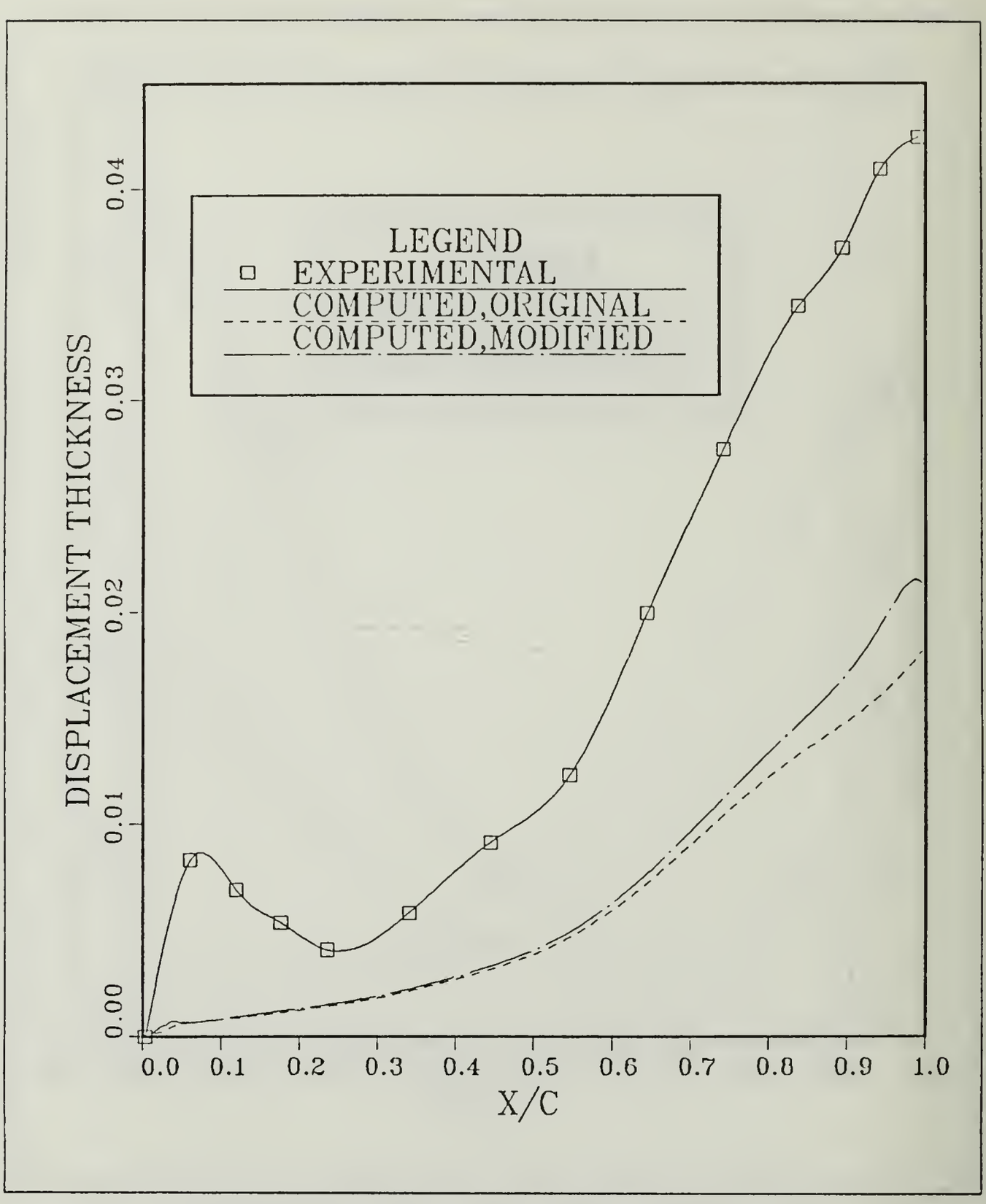

Figure 18. Displacement thickness on the upper surface (β = 46°)

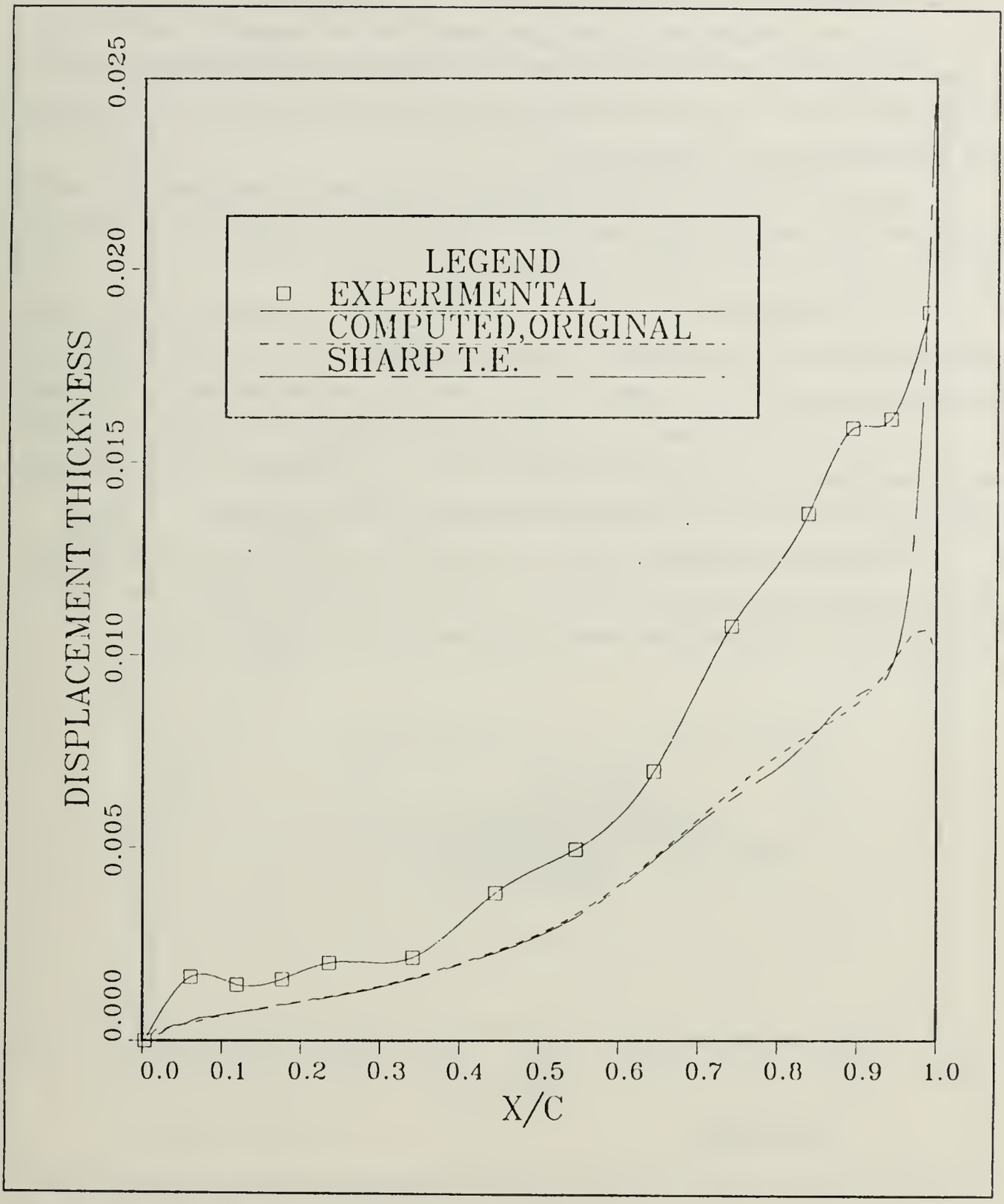

Figure 19. The effect of sharp trailing edge.

4. Comparison to a Navier Stokes Code

A limited comparison of the experimental and the computed results with a Navier Stokes (N.S.) code calculations has been performed. The N.S. code has been developed and run by S. J. Shamroth of Scientific Research Associates inc. in cooperation with Pratt and Whitney Aircraft.

Since the N.S. code does not compute the displacement thickness, the velocity profiles near the surface of the blade were compared. The comparisons were made at 90% chord on the suction surface for all three inlet angles.

At the design point, $\beta = 40°$, shown in Figure 20 on page 43, both the interactive code and the N.S. code failed to predict accurately the actual velocity profile. In this case the interactive code seems to yield somewhat better results than the N.S. code.

At the higher inlet angles, $\beta = 43.4°$ and $\beta = 46°$, shown in Figure 21 on page 44 and in Figure 22 on page 45 respectively, the N.S. calculations show significantly better agreement with the experimental results than the interactive code.

From these comparisons, it can be seen that the interactive code deviation from the actual results increases with increased inlet angle (increased loading of the cascade), whereas the N.S. code deviation seems to decrease with increased inlet angle.

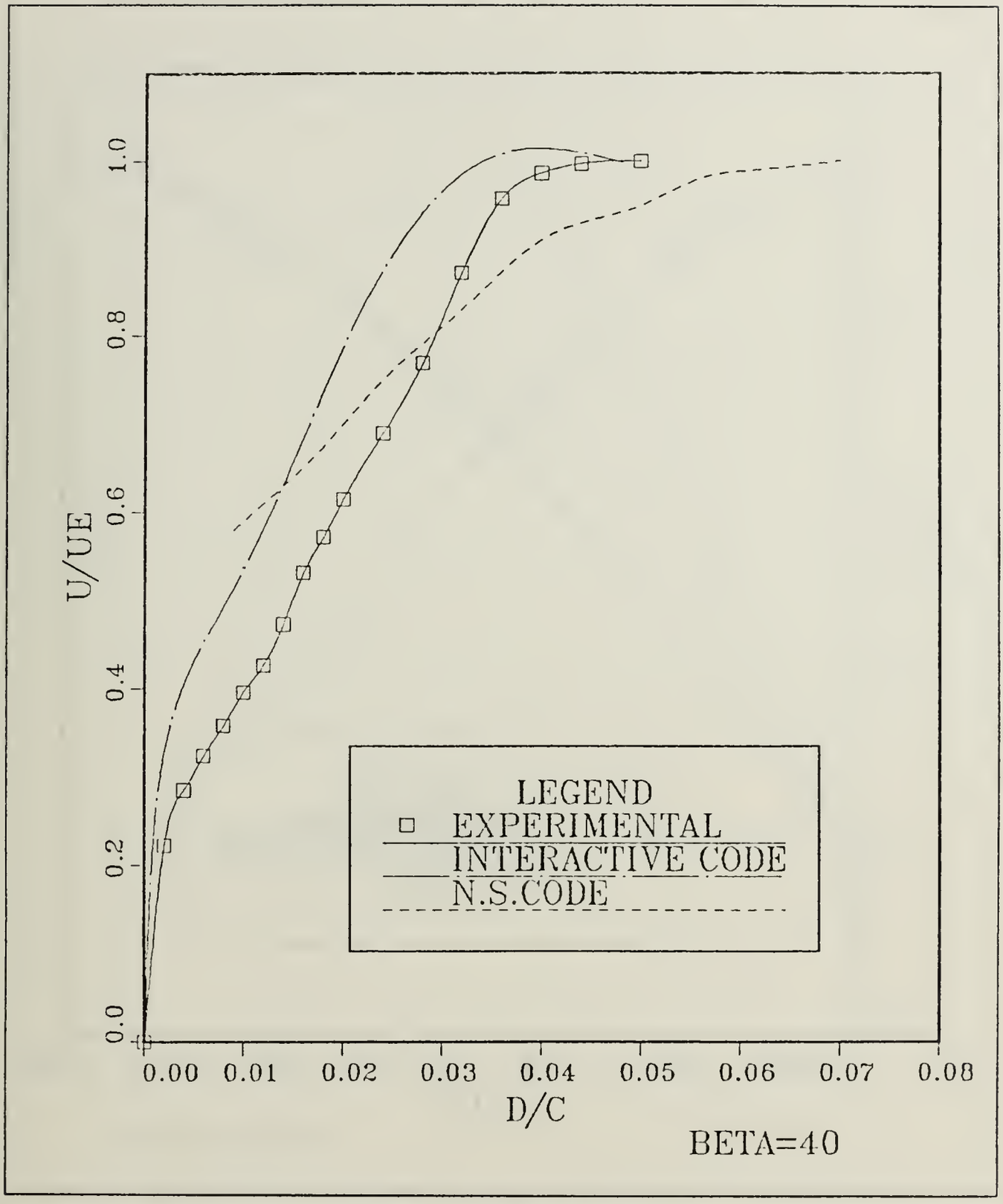

Figure 20. The results of the N. S. code at $\beta = 40°$

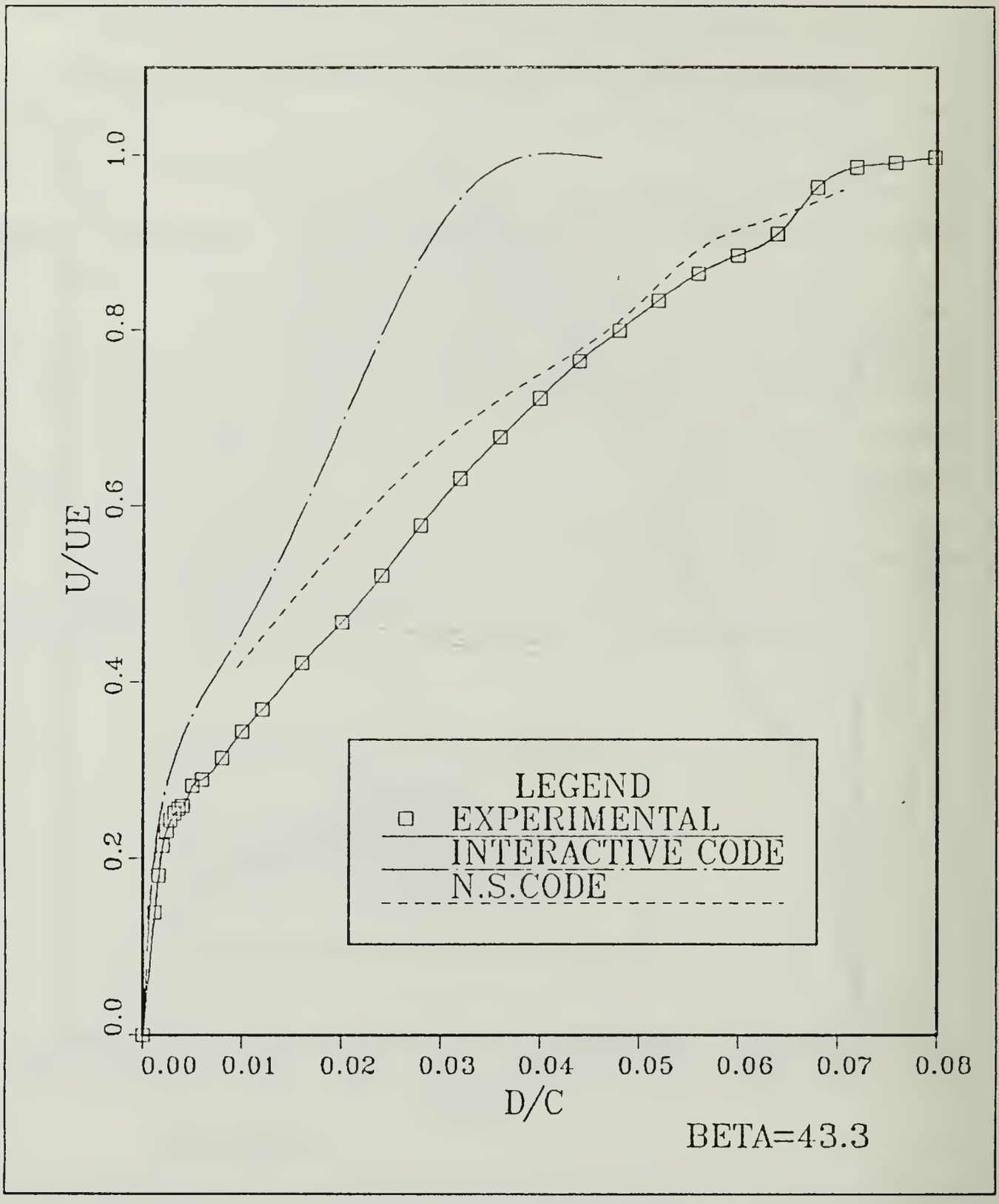

Figure 21. The results of the N. S. code at $\beta = 43.4°$

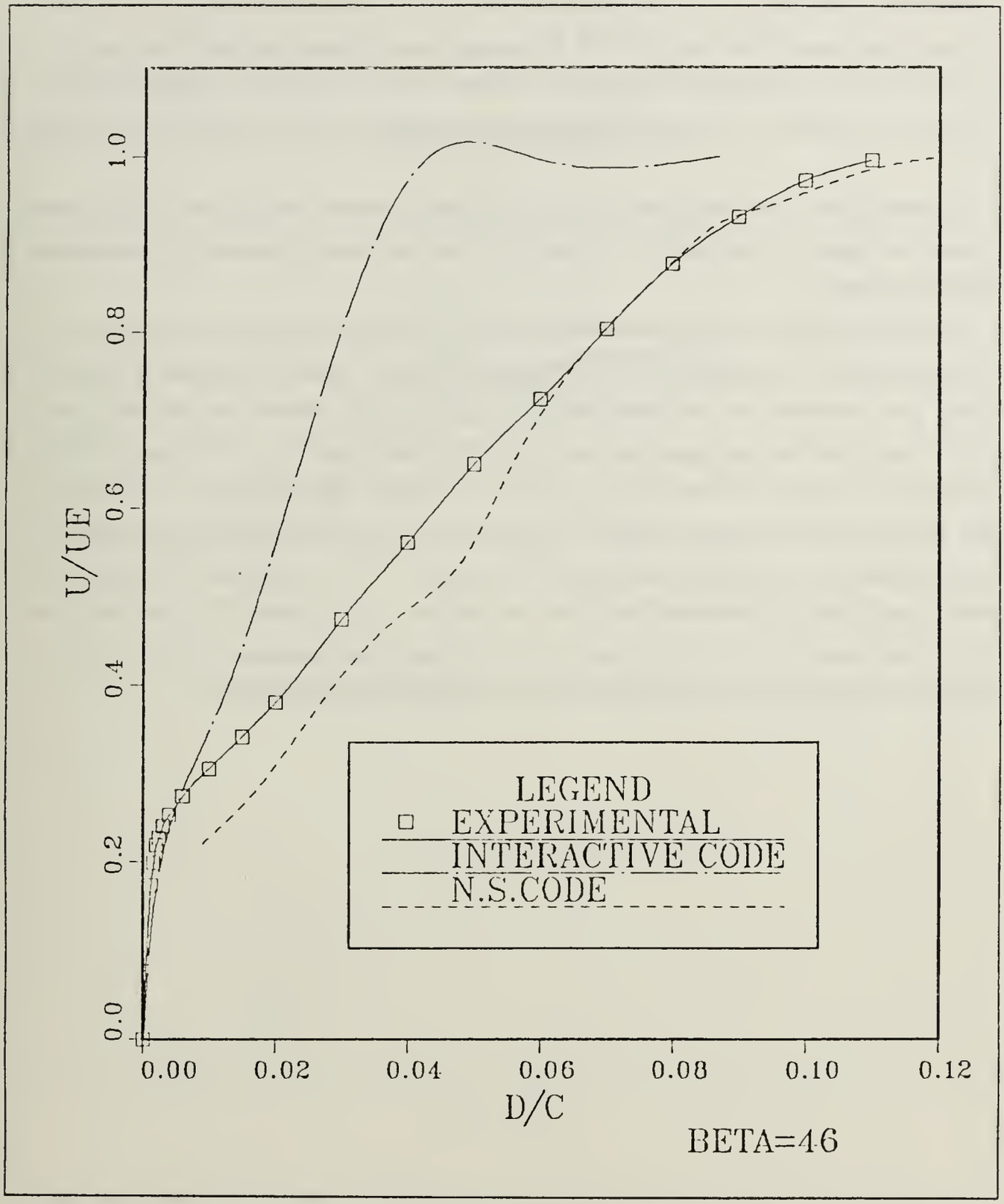

Figure 22. The results of the N. S. code at $\beta = 46°$

B. P & W CASCADE

The experimental data for the P & W cascade was obtained at inlet flow angle of 52°, at M = 0.11, and Reynolds number of 478000. The cascade had a stagger angle of 15.75° and 0.7 spacing. A general layout of the cascade is shown in Figure 23 on page 47.

A comparison of the computed and the measured pressure coefficients on the blade is shown in Figure 24 on page 48. There is a good agreement between the computed and the measured C_p.

The displacement thickness was measured in the experiment only at 96.8% of chord. This measurement is compared to the computed results in Figure 25 on page 49. As can be seen, the computed and the measured data agree almost perfectly on the lower surface, and quite well on the upper surface. The difference observed on the upper surface is caused by the early prediction, by the code, of trailing edge separation, a short distance upstream of the actual location. This can also be observed when comparing the velocity profiles at that point, in Figure 26 on page 50. The computed velocity curve shows a small zone of reversed flow near the surface of the blade. This reversed flow could be the result of a too early prediction of trailing edge separation by the code, or it could have existed in the actual flow but not detected because of its size.

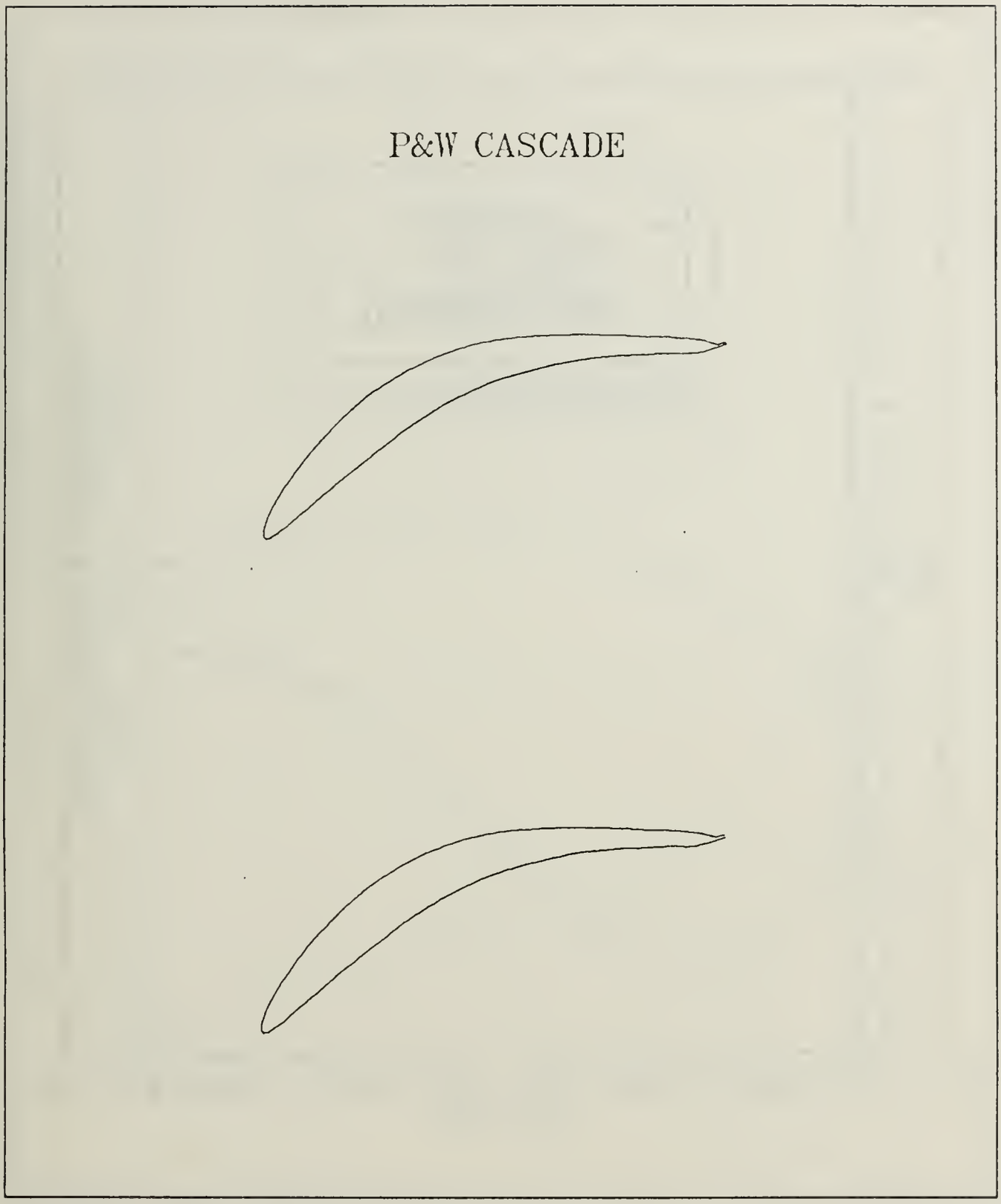

Figure 23. Pratt & Whiteny cascade

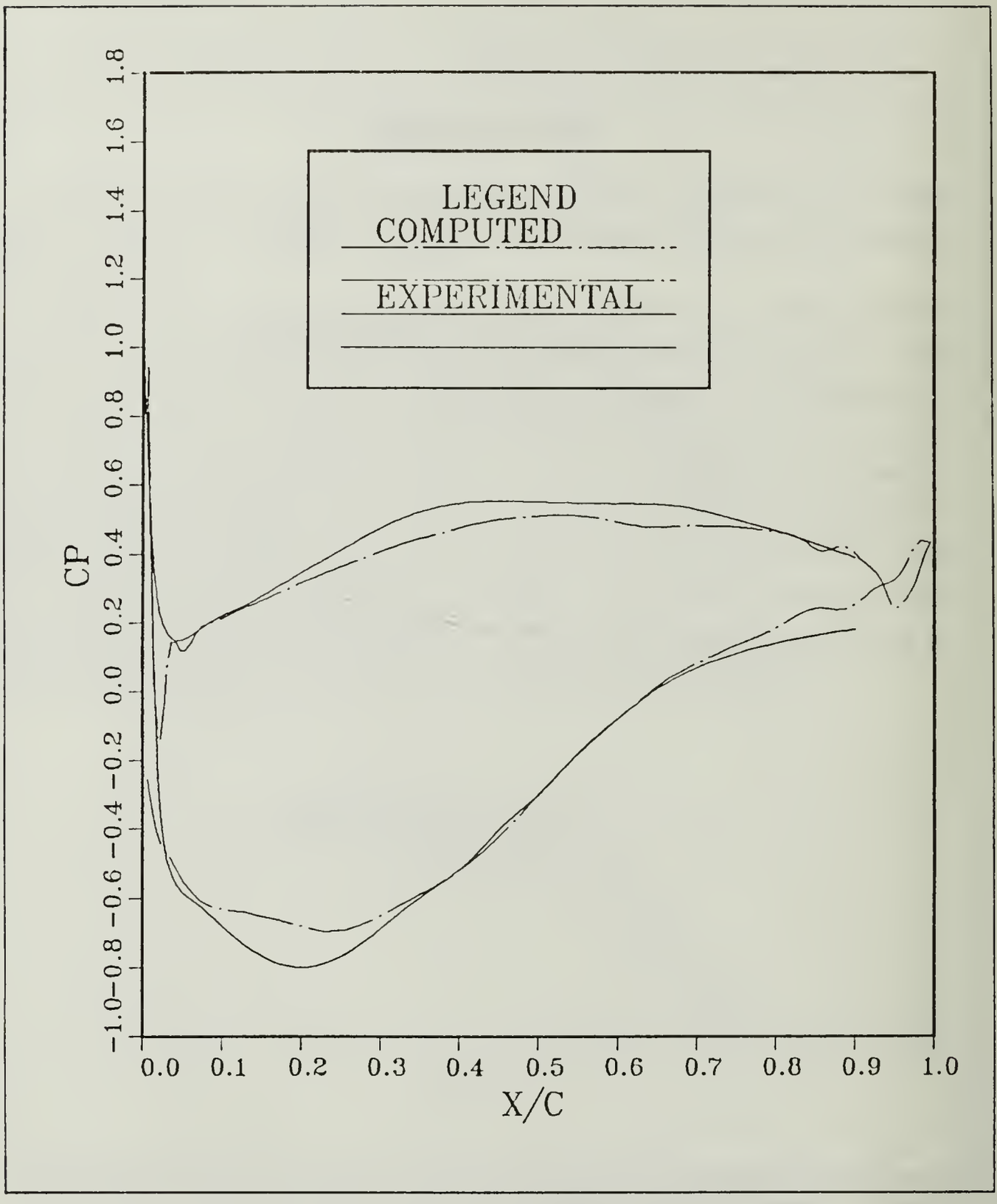

Figure 24. Comparison of pressure coefficient.

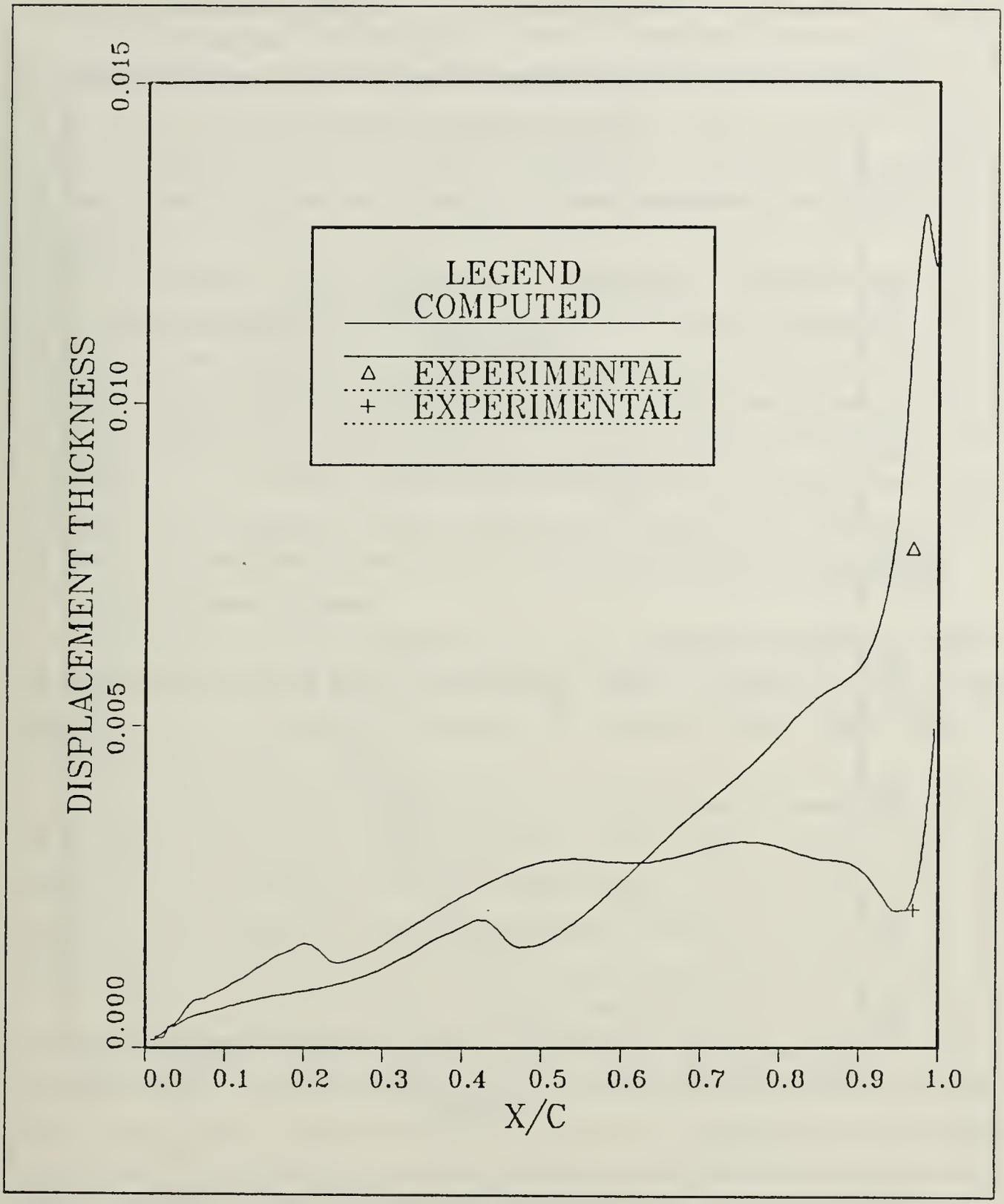

Figure 25. **Displacement thickness comparison:** Experimental data shown at 96.8% chord.

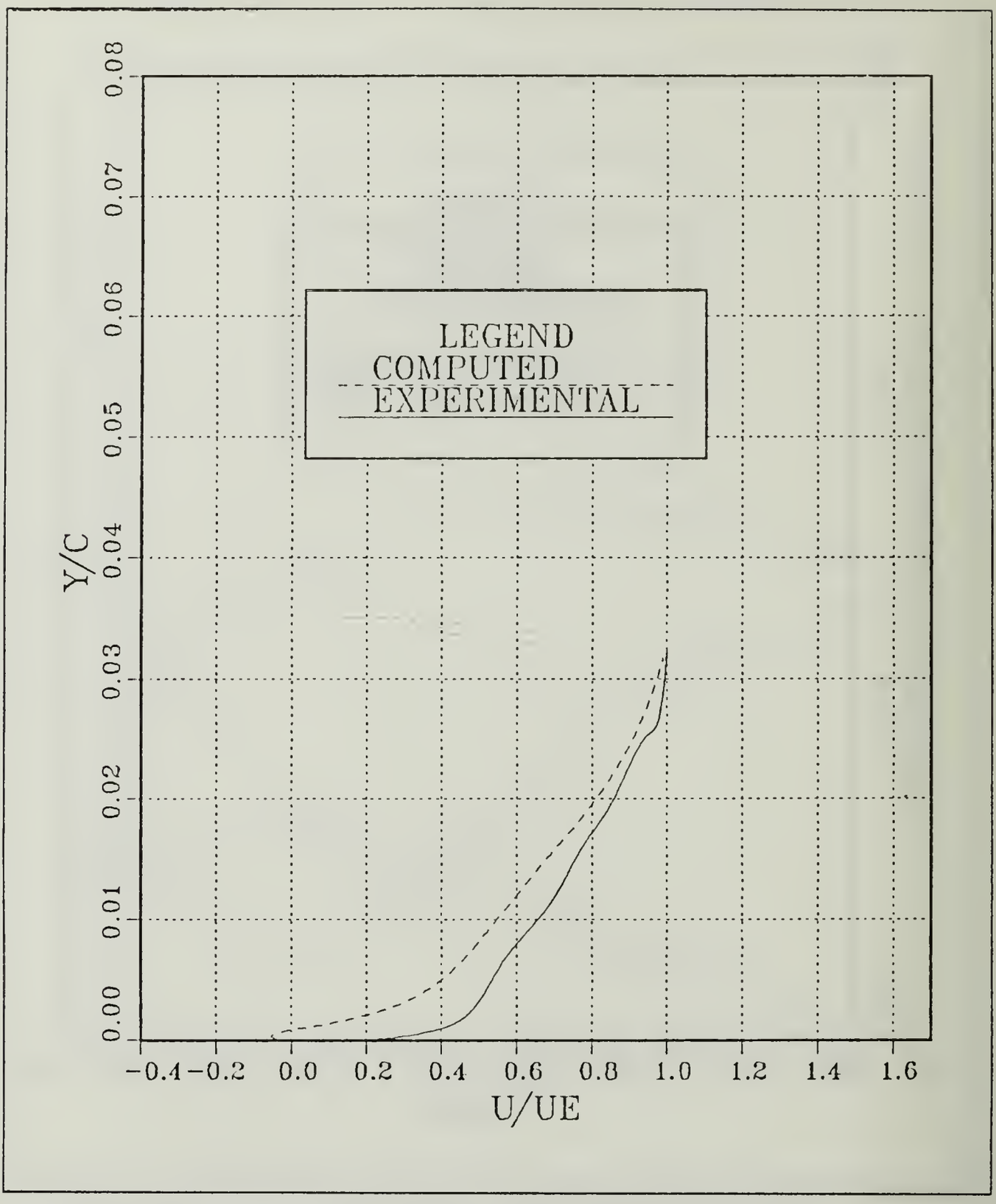

Figure 26. Velocity profile at 96.8% chord on the upper surface.

C. C4 CASCADE

The C4 cascade has a stagger angle of 29.5°, a camber angle of 31.1° and spacing of 0.992. It has been tested at Reynolds numbers of about 200000, and inlet angles of 34.1° to 47.7° (which corresponds to incidence angles of -10.9° to 2.7°). The general layout of the cascade is shown in Figure 27 on page 52. A computer code that generates the coordinates of the blade and a summary of some experimental results are given in Appendix B.

The code was run with the intermittency constant $G_\gamma = 10$. Higher values of G_γ (above 100) caused numerical problems in the code. The onset of transition was first taken at the point where it was observed in the experiment. At the lower inlet angle it seems that a better agreement with the experimental results can be obtained by delaying the onset of transition but trying to implement it resulted in numerical breakdown of the computation. At the higher inlet angles, better agreement with the experimental results was achieved by initiating the transition earlier (at 26% chord for $\beta = 45.6°$ and at 21% for $\beta = 47.7°$ as compared to 44% and 36% chord as observed in the experiment).

1. Displacement Thickness

Comparisons of the experimental data to the computed displacement thickness are shown in Figure 28 on page 53 for inlet angle of 34.1°, in Figure 29 on page 54 for inlet angle of 36.3°, in Figure 30 on page 55 for inlet angle of 45.6° and in Figure 31 on page 56 for inlet angle of 47.7°.

As can be seen in the figures, there is a good agreement between the actual and the computed results at the two lower angles ($\beta = 34.1°$ and $\beta = 36.3°$, in which the incidence angles were negative). At the two higher angles, $\beta = 45.6°$ and $\beta = 47.7°$ the computed results agree with the actual results up to about 70% chord, and then the displacement thickness predicted by the code becomes much thicker than the actual one.

The code predicted a large flow separation area starting at about 70% chord at the lower inlet angles, and at about 46% chord at the higher inlet angles. This flow separation was not observed in the experiment. The discrepancies between the computed and the actual results behind 60% to 70% chord can be explained by the inaccurate calculations by the code due to the large separated areas. When the code encounters separation, several approximations are made (like the FLARE approximation) based on the assumption that the separated area is small. When the separated area is large, these approximations may result in inaccurate prediction of the flow field.

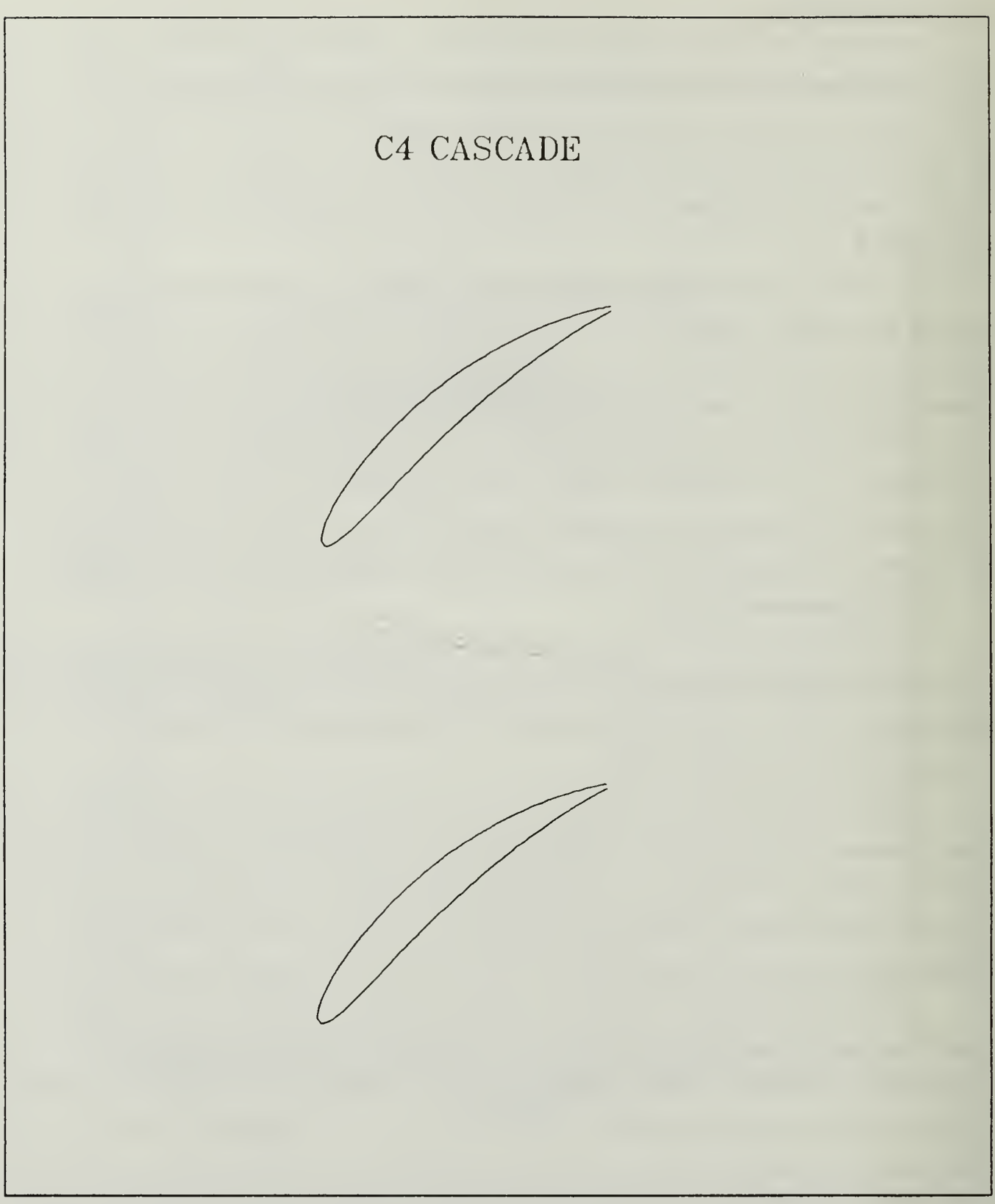

Figure 27. C4 Cascade

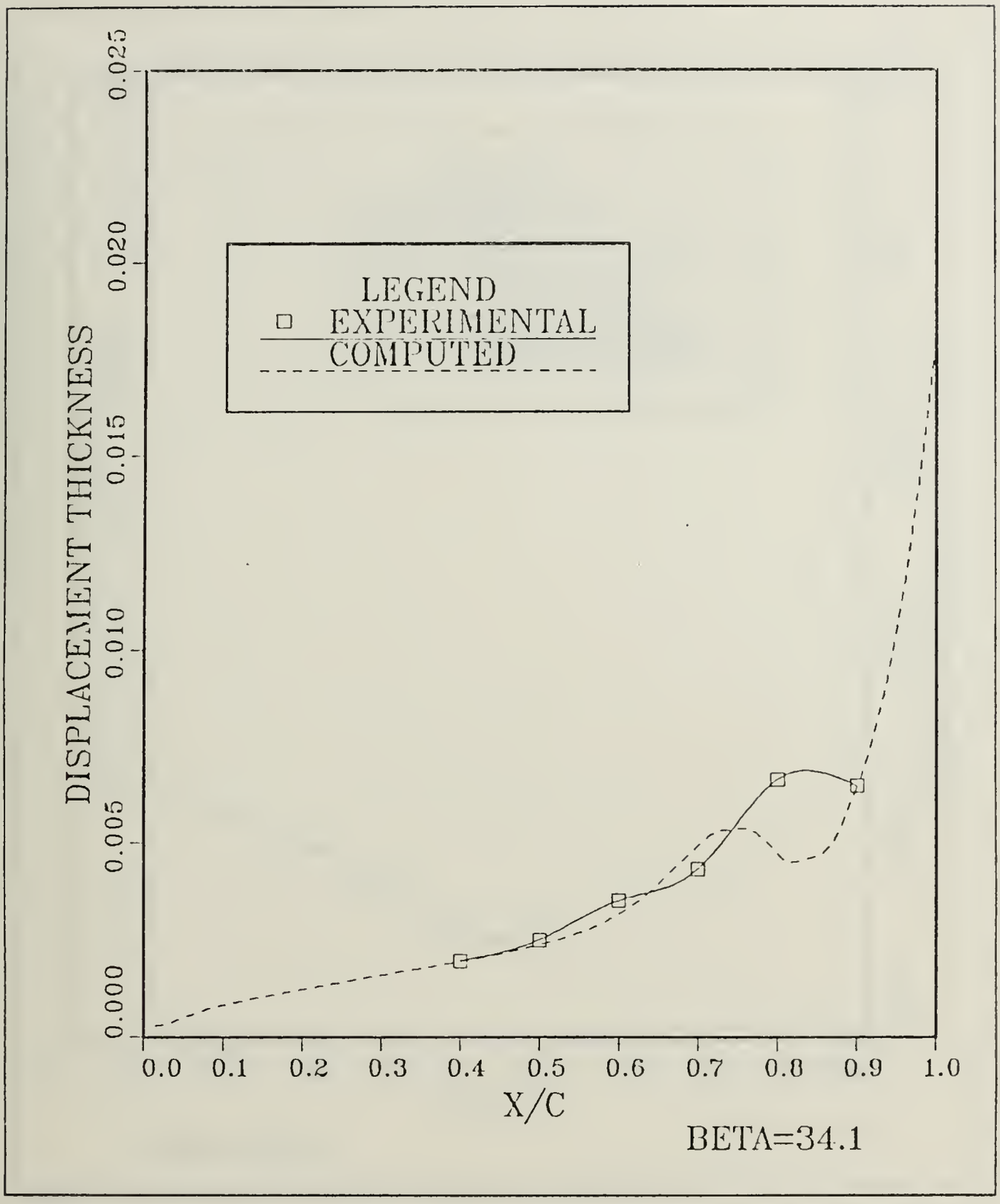

Figure 28. C4 cascade at $\beta = 34.1°$: Displacement thickness comparison with computed results.

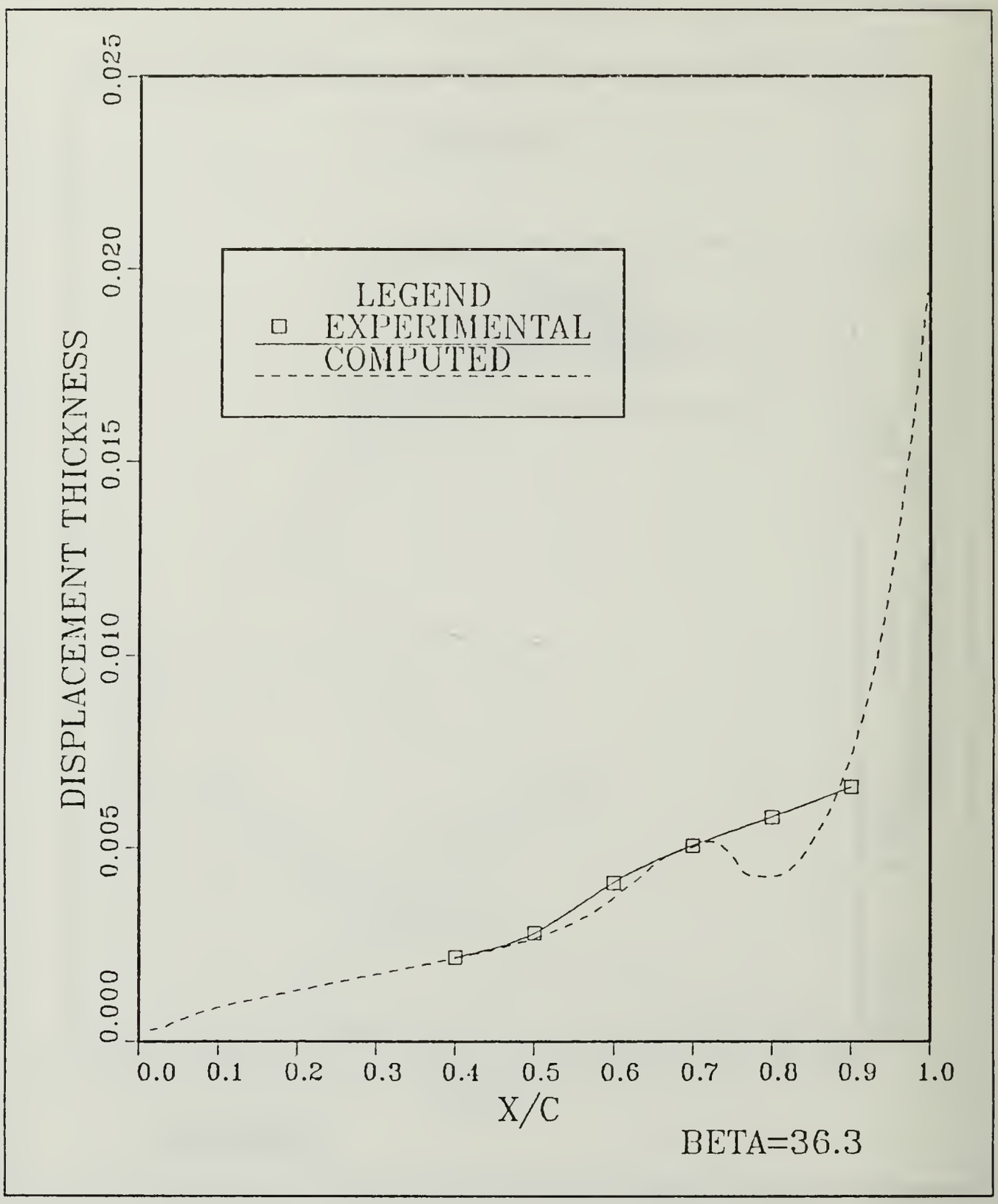

Figure 29. C4 cascade at β = 36.3°: Displacement thickness comparison with computed results.

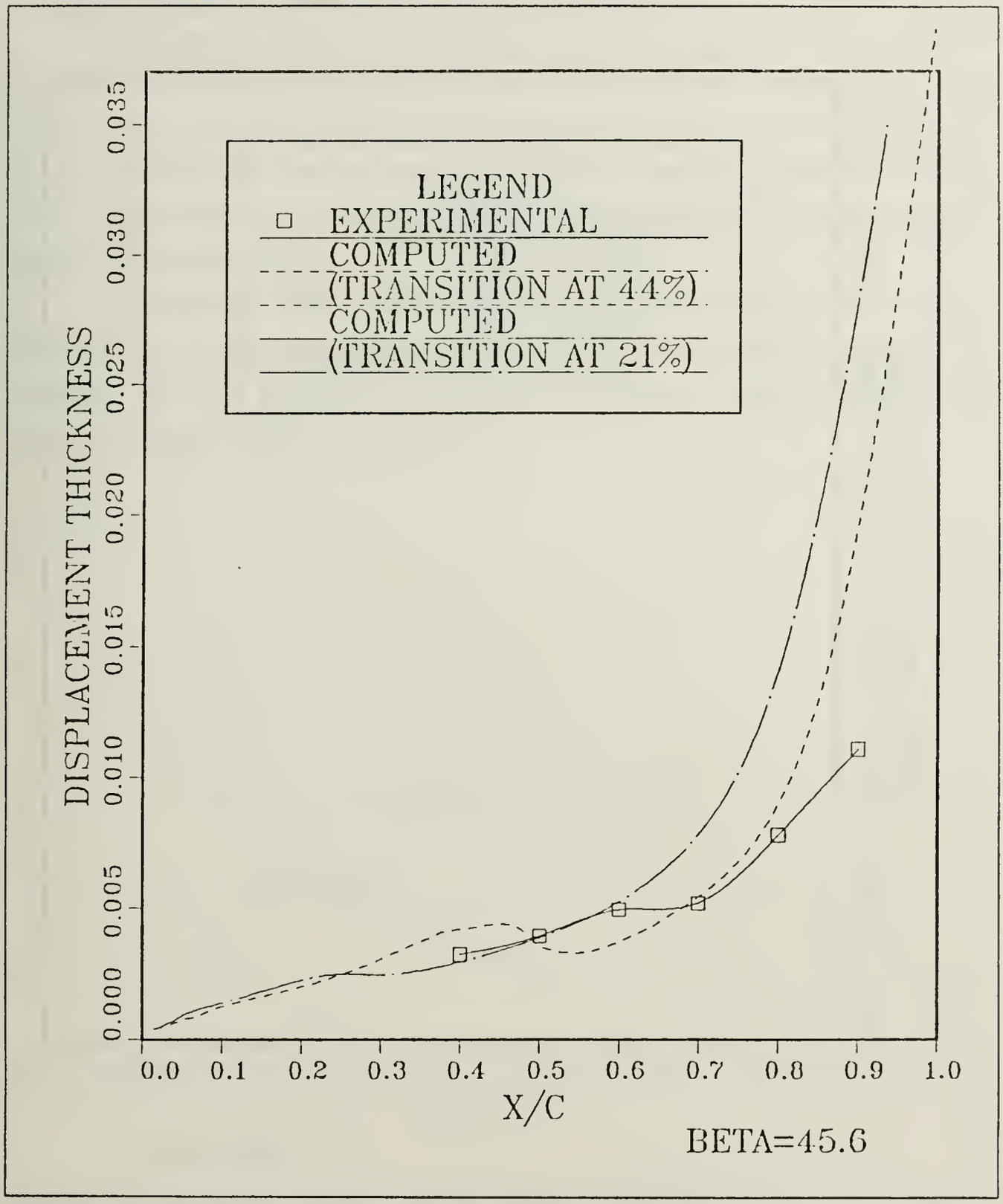

Figure 30. C4 cascade at $\beta = 45.6°$: Displacement thickness comparison with computed results (G = 10).

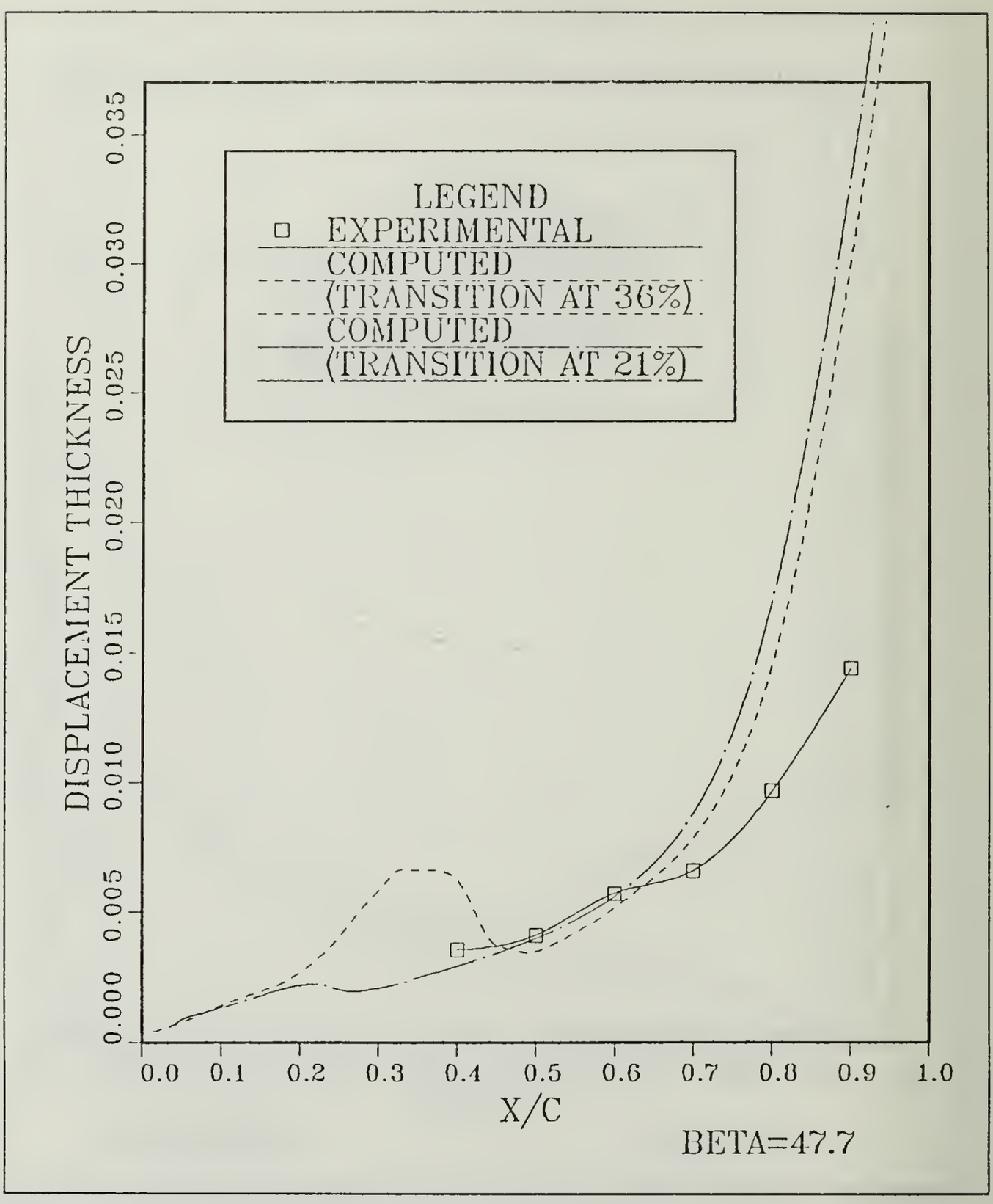

Figure 31. C4 cascade at β = 47.7°: Displacement thickness comparison with computed results (G = 10).

2. External Velocity and Velocity Profiles Comparisons

A comparison of the external velocity on the upper surface of the blade is shown in Figure 32 on page 58 for inlet angle of 45.6° and in Figure 33 on page 59 for inlet angle of 47.7°. It can be seen that there is a good agreement between the experimental and the computed results up to about 80% chord. Near the trailing edge the computed results deviate from the experimental results due to the inaccuracy in the calculations of the displacement thickness.

A comparison of the velocity profiles in the boundary layer at 50% chord is shown in Figure 34 on page 60 for inlet angle of 34.1° and in Figure 35 on page 61 for inlet angle of 36.3°. The agreement between the calculated velocity profiles and the measured velocity profiles is very good.

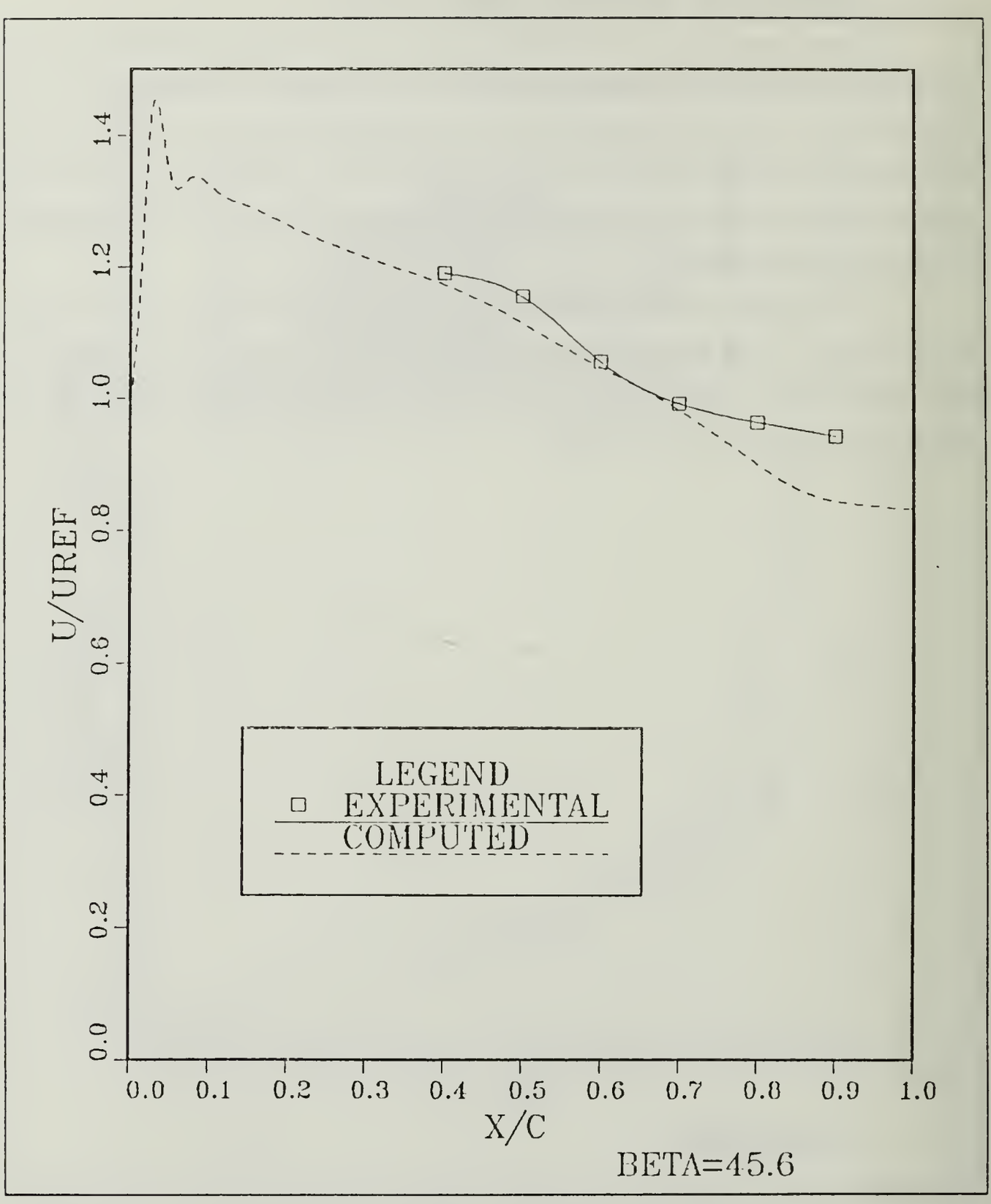

Figure 32. C4 cascade at β = 45.6°: External velocity distribution.

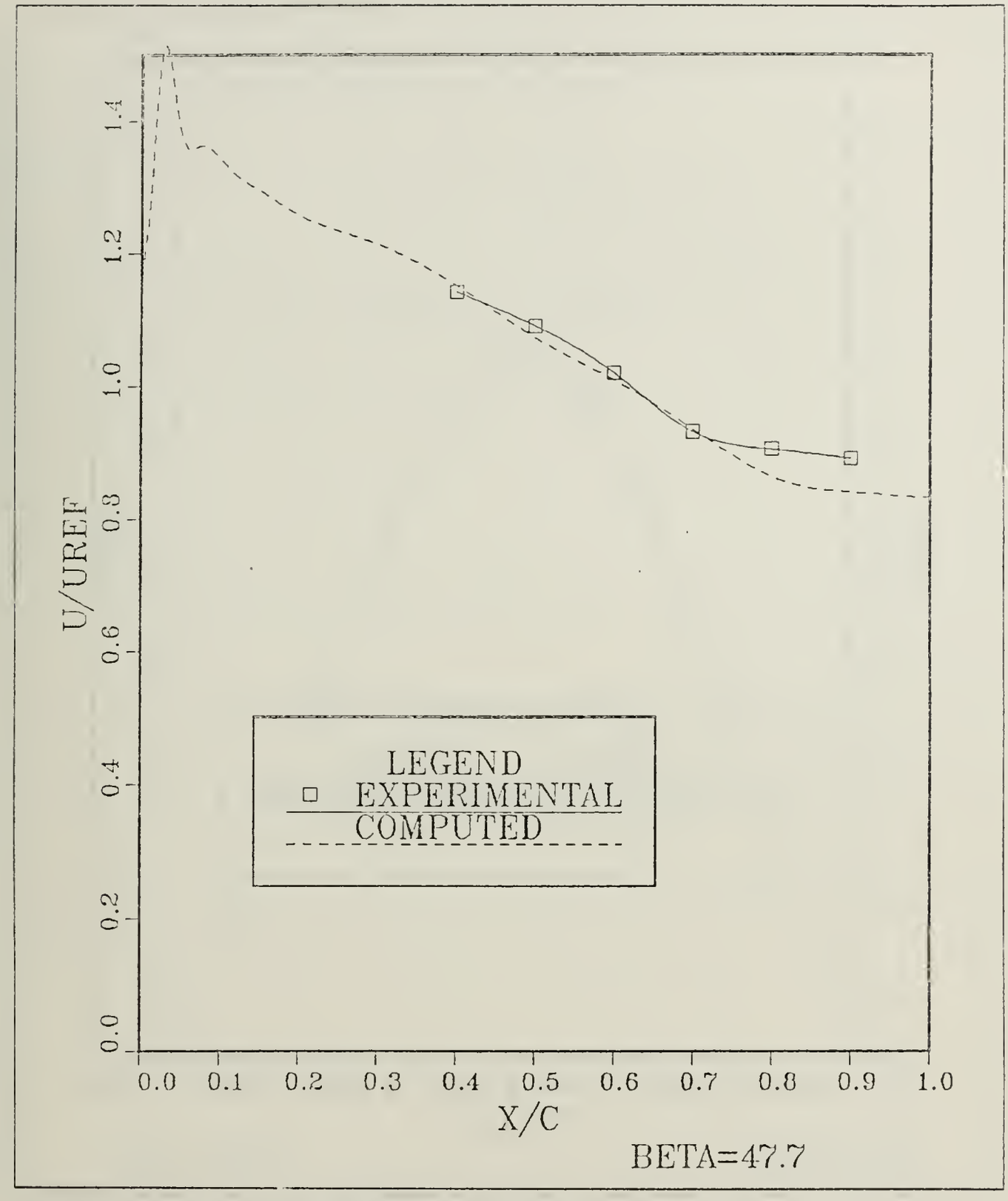

Figure 33. C4 cascade at $\beta = 47.7°$: External velocity distribution.

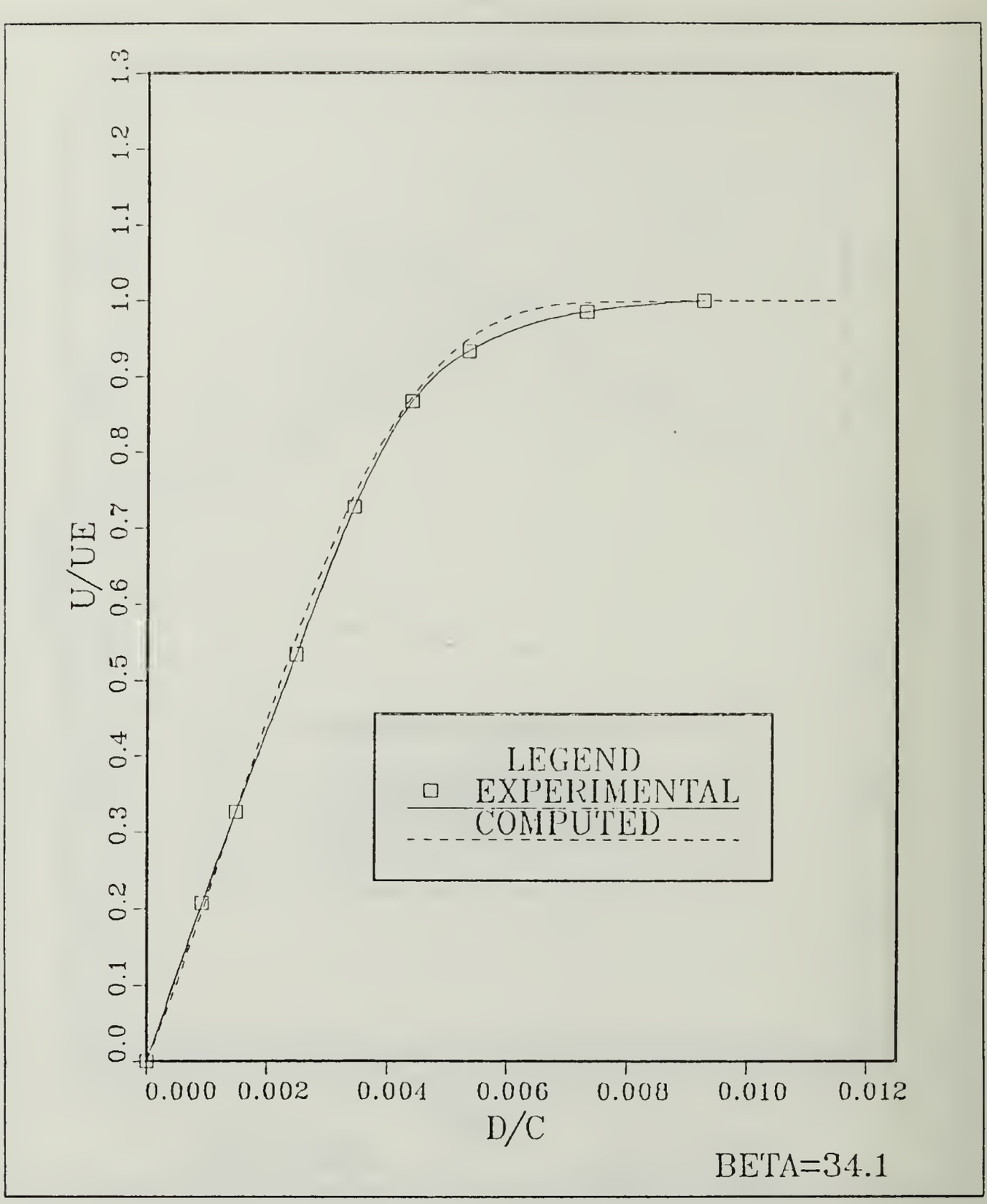

Figure 34. C4 cascade at $\beta = 34.1°$: Velocity profile at 50% chord.

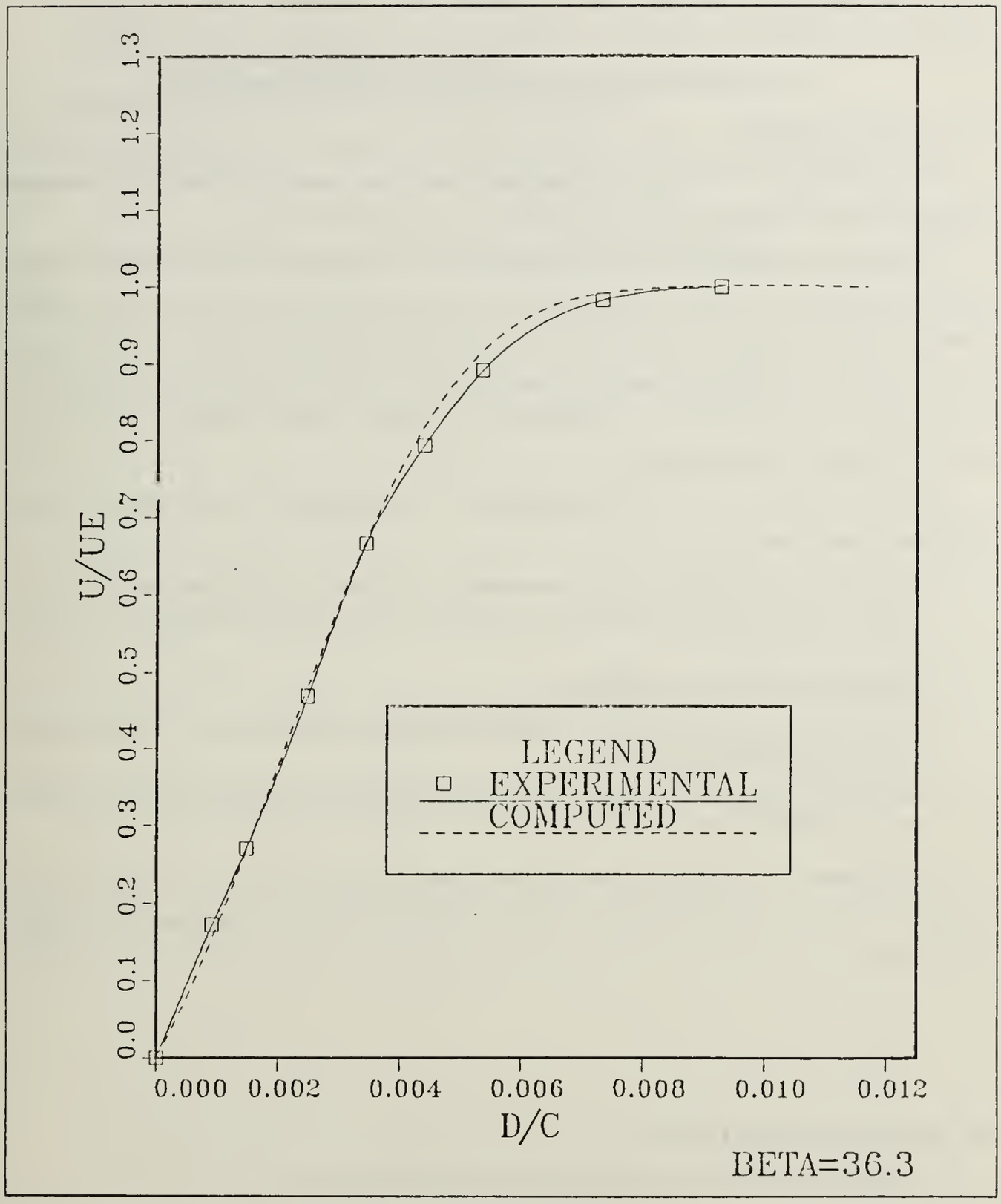

Figure 35. C4 cascade at $\beta = 36.3°$: Velocity profile at 50% chord.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The interactive viscous inviscid computer code, has been investigated by comparing its predictions of boundary layer parameters to experimental data.

It has been found that the code yields reasonable results for lightly loaded cascades, but the prediction of the boundary layer thickness on the suction surface of highly loaded cascades deviates significantly from experimentally measured data. In two cases involving highly cambered cascade blades, with sharp leading edge, the code failed to run. It was also found that the prediction of external velocity distribution on highly loaded cascades was inaccurate.

The main reasons to the discrepancies in the prediction of the boundary layer thickness seem to be:

1. Inaccuracy in predicting flow parameters in regions of large flow separation (due to inadequate transition model and approximations made in calculating the flow in separated areas).
2. Inaccurate turbulence modelling.
3. Possible violation of the basic assumptions of the boundary layer theory in areas of very thick boundary layers.
4. The wake is not calculated by the code. The result is inaccurate flow prediction near the trailing edge.

The inaccuracy in the prediction of the external velocity distribution in highly loaded cascades is due to the interaction law, which does not account for the presence of adjacent blades.

B. RECOMMENDATIONS

The recommended steps in order to improve the code are:

1. Improving the interaction law by assuming a distribution of sources on the actual surface (instead of the assumption of a flat plate), letting the correction term to the external velocity vary across the boundary layer and distributing sources on the adjacent blade as well for better modelling of the boundary layer effect on the external velocity.

2. Changes to the derivation of the boundary layer equations should be investigated to allow a better treatment of thick boundary layers (like omitting the assumption of $\partial p/\partial y = 0$ across the boundary layer).
3. Different turbulence models should be investigated.
4. The wake should be included in the calculations.

APPENDIX A. COMPUTER CODE LISTING

```
C*************************************************************************************INT0001
C******   VISCOUS-INVISCID INTERACTION PROGRAM FOR CASCADE FLOWS     ******INT0002
C*************************************************************************************INT0003
C                                                                                     INT0004
C                        V E R S I O N   3.A                                          INT0005
C                            JANUARY 87                                               INT0006
C                                                                                     INT0007
C   THIS VISCOUS-INVISCID INTERACTION METHOD, CAPABLE OF COMPUTING BOTH               INT0008
C   SINGLE AIRFOIL AND CASCADE FLOWS, WAS DEVELOPED BY CEBECI AND                     INT0009
C   COLLABORATEURS AT LONG BEACH STATE AND DOUGLAS AIRCRAFT COMPANY.                  INT0010
C   THE CODE APPLIES TO INCOMPRESSIBLE, 2-DIMENSIONAL, STEADY FLOWS                  INT0011
C   PAST LINEAR, ARBITRARILY STAGGERED CASCADES. THE METHODS BASIC                    INT0012
C   INGREDIENTS INCLUDE                                                             INT0013
C   1.  A FIRST ORDER PANEL METHOD TO SOLVE LAPLACE'S EQUATION,                       INT0014
C   2.  A FINITE DIFFERENCE SCHEME TO SOLVE THE BOUNDARY LAYER EQUATIONS             INT0015
C       SUBJECT TO DIRECT OR INTERACTIVE BOUNDARY CONDITIONS,                       INT0016
C   3.  A STRONG INTERACTION MODEL TO COUPLE VISCOUS AND INVISCID FLOW               INT0017
C       RESULTS, AND                                                                INT0018
C   4.  A ZERO EQUATION, ALGEBRAIC TURBULENCE MODEL TO ESTIMATE                     INT0019
C       TURBULENT SHEAR STRESSES.                                                   INT0020
C                                                                                   INT0021
C   IN SUMMARY, THE CODE WILL PROVIDE, FOR ATTACHED AS WELL AS MODERATE-            INT0022
C   LY SEPARATED FLOWS PAST SINGLE AIRFOILS OR CASCADES, THE FOLLOWING              INT0023
C   1.  INVISCID AND VISCOUS PRESSURE DISTRIBUTIONS,                                INT0024
C   2.  DISTRIBUTIONS OF                                                            INT0025
C       A.  LOCAL SKIN FRICTION COEFFICIENT,                                        INT0026
C       B.  DISPLACEMENT AND MOMENTUM THICKNESS, AND                                INT0027
C   3.  VELOCITY PROFILES ACROSS THE BOUNDARY LAYER.                                INT0028
C                                                                                   INT0029
C   MODIFICATIONS SINCE VERSION 3.0:                                                INT0030
C   1.  PRECISE ASSIGNMENT OF BEGIN OF TRANSITION.                                  INT0031
C   2.  CORRECTION OF AN ERROR IN THE CALCULATION OF MOMENTUM THICKNESS.            INT0032
C   3.  ADDITIONAL PRINT OPTION: IP=-2 WILL PROVIDE AN INPUT FILE (UNIT             INT0033
C       NUMBER 12) FOR THE PLOTTING ROUTINE.                                        INT0034
C                                                                                   INT0035
C                                                                                   INT0036
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                                INT0037
      COMMON/BLOW/VN(100)                                                           INT0038
      COMMON/BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),                    INT0039
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)                   INT0040
      COMMON/CASCDE/INLET,SP,SINGLE,ALPHAA,ALPHAI,STAG                              INT0041
      COMMON/TRN/ PGAMTR,OMEGA,RTHETB,RTRANB                                        INT0042
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                             INT0043
      DIMENSION XO(100),YO(100),X(100),Y(100),VCOM(100),DLS(100),                   INT0044
     + XS(100),YS(100),XSTGR(100),YSTGR(100),DBPP(100)                              INT0045
      DIMENSION CASEID(20),XCTRI(2),ITRI(2),NBL(2)                                  INT0046
      LOGICAL SINGLE,TRFIND                                                         INT0047
      TRFIND(1)= .FALSE.                                                            INT0048
      TRFIND(2)= .FALSE.                                                            INT0049
```

```
      ICASE  = 0                                                          INT00500
1     READ(5,5,END=999) TITLE                                             INT00510
5     FORMAT(20A4)                                                        INT00520
      ICASE = ICASE + 1                                                   INT00530
      REWIND 3                                                            INT00540
C     READ (5,10)                                                         INT00550
10    FORMAT(1X)                                                          INT00560
      READ (5,20) ITRI(1),ITRI(2),IRST,ICYTL,IP                           INT00570
20    FORMAT(16I5)                                                        INT00580
C      READ (5,10)                                                        INT00590
      READ (5,25) INLET,ISTAG,ALPHAI,STAG,SP,PGAMTR,OMEGA                 INT00600
25    FORMAT(2I5,5F10.0)                                                  INT00610
      READ (5,27)RN,XCTRI(1),XCTRI(2),ALPHAA                              INT00620
27    FORMAT(4E10.0)                                                      INT00630
      IF (IP.EQ.-2) THEN                                                  INT00640
         READ (5,20) NVP(1),NVP(2)                                        INT00650
         IF (NVP(1).NE.0) READ (5,20) (NXVP(I,1),I=1,NVP(1))              INT00660
         IF (NVP(2).NE.0) READ (5,20) (NXVP(I,2),I=1,NVP(2))              INT00670
      END IF                                                              INT00680
      IF (ICASE .EQ. 1) READ (5,20) N,NI                                  INT00690
      IREAD = 1                                                           INT00700
      IBLOW = 1                                                           INT00710
      SINGLE = .FALSE.                                                    INT00720
      IF (SP .LE. 0.0) SINGLE = .TRUE.                                    INT00730
      N = N - 1                                                           INT00740
      N1= N + 1                                                           INT00750
      IF (ICASE .GT. 1) THEN                                              INT00760
      N1 = N1SAVE                                                         INT00770
      N = N - 1                                                           INT00780
      GOTO 53                                                             INT00790
      END IF                                                              INT00800
      IF (IREAD .EQ. 1) GO TO 40                                          INT00810
C      READ (5,10)                                                        INT00820
      READ (5,30) (XO(I), YO(I), I=1,N+1)                                 INT00830
30    FORMAT(2F10.0)                                                      INT00840
      GO TO 50                                                            INT00850
C                                                                         INT00860
C40    READ(5,10)                                                         INT00870
40    READ(5,45) (XO(I) , I=1,N+1)                                        INT00880
C      READ(5,10)                                                         INT00890
      READ(5,45) (YO(I) , I=1,N+1)                                        INT00900
45    FORMAT(6F10.0)                                                      INT00910
C                                                                         INT00920
50    CONTINUE                                                            INT00930
      IF (IP.EQ.-2) THEN                                                  INT00940
         WRITE(12,20) N+1,NVP(1),NVP(2),90,70,INLET                       INT00950
         IF (NVP(1).NE.0) WRITE(12,20) (NXVP(I,1),I=1,NVP(1))             INT00960
         IF (NVP(2).NE.0) WRITE(12,20) (NXVP(I,2),I=1,NVP(2))             INT00970
         IF (INLET.NE.1) WRITE(12,80) RN,ALPHAA                           INT00980
         IF (INLET.EQ.1) WRITE(12,80) RN,ALPHAI                           INT00990
         WRITE(12,82) (XO(I),I=1,N+1)                                     INT01000
         WRITE(12,82) (YO(I),I=1,N+1)                                     INT01010
   80    FORMAT(2E15.5)                                                   INT01020
   82    FORMAT(8F10.6)                                                   INT01030
      END IF                                                              INT01040
C                                                                         INT01050
```

```
      NRITE = (NI+1)/2                                                  INT01060
      IMIN = (N1-1)/2+1                                                 INT01070
      IF((N1/2*2) .EQ. N1) IMIN = N1/2                                  INT01080
      CALL TRGRID (N1,XO,YO,NI,NRITE,0.5,IMIN,RAD,1,NXSS1)              INT01090
      N1SAVE = N1                                                       INT01100
53    CONTINUE                                                          INT01110
      ALPHAA = 0.0174533 * ALPHAA                                       INT01120
      ALPHAI = 0.0174533 * ALPHAI                                       INT01130
      STAG   = 0.0174533 * STAG                                         INT01140
C                                                                       INT01150
      IF (INLET .EQ. 0) THEN                                            INT01160
      ALPHA = ALPHAA                                                    INT01170
      ELSE                                                              INT01180
      ALPHA = ALPHAI                                                    INT01190
      END IF                                                            INT01200
C                                                                       INT01210
      IF (ISTAG .NE. 0) THEN                                            INT01220
      CALL STAGR(N1,STAG,XO,YO,XSTGR,YSTGR)                             INT01230
      ELSE                                                              INT01240
      DO 55 I = 1 , N1                                                  INT01250
      XSTGR(I) = XO(I)                                                  INT01260
      YSTGR(I) = YO(I)                                                  INT01270
55    CONTINUE                                                          INT01280
      END IF                                                            INT01290
C                                                                       INT01300
C  READ DATA FROM VISCOUS CAL.                                          INT01310
C                                                                       INT01320
      ICYCLE = 0                                                        INT01330
60    ICYCLE = ICYCLE + 1                                               INT01340
C                                                                       INT01350
      CALL POTNL(N1,IRST,ALPHA,CHORD,XO,YO,XSTGR,YSTGR,X,Y,DLS,VCOM,    INT01360
     +           DBPP)                                                  INT01370
      IF (ICYCLE .GT. ICYTL) THEN                                       INT01380
      REWIND 3                                                          INT01390
      WRITE (3)N1,(XO(I),YO(I),DLS(I),VN(I),DBPP(I),I=1,N1)             INT01400
      GOTO 1                                                            INT01410
      END IF                                                            INT01420
C                                                                       INT01430
      IF (ISTAG .NE. 0) THEN                                            INT01440
      DO 70 I = 1 , N1-1                                                INT01450
      X(I) = 0.5 * (XO(I)+XO(I+1))                                      INT01460
      Y(I) = 0.5 * (YO(I)+YO(I+1))                                      INT01470
70    CONTINUE                                                          INT01480
      END IF                                                            INT01490
C                                                                       INT01500
      CALL CASBLP(N1,XO,YO,X,Y,XS,YS,DLS,VCOM,DBPP,RN                   INT01510
     +            ,NBL,ITRI,XCTRI,TITLE)                                INT01520
      GO TO 60                                                          INT01530
999   CONTINUE                                                          INT01540
      STOP                                                              INT01550
      END                                                               INT01560
C                                                                       INT01570
      SUBROUTINE POTNL(N1,IRST,ALPHA,CHORD,XO,YO,XSTGR,YSTGR,X,Y,DLS,   INT01580
     +                 VCOM,DBPP)                                       INT01590
C                                                                       INT01600
      COMMON/BLOW/VN(100)                                               INT01610
```

```
      COMMON /BLCO/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT01620
      COMMON/BLIN/TITLE(20),XC(100),YC(100),ISG(100),DELS(100),         INT01630
     +            XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)        INT01640
      COMMON/CASCDE/ INLET,SP,SINGLE,ALPHAA,ALPHAI,STAG                 INT01650
C  SIMPLE SOURCE POTENTIAL CODE                                         INT01660
      DIMENSION AOFF(100,100), BOFF(100,100), XP(100), YP(100),X(100),  INT01670
     +     S(100),C(100), D(100),VTAN(3,100),VNOR(3,100),R(3,100),      INT01680
     +     VCOM(100),SIGCOM(100),CP(100),XO(100),YO(100)                INT01690
     +    ,VNC(100),D1(100),D2(100),D3(100),SO(100),SC(100)             INT01700
     +    ,XOFF(100),YOFF(100),T(3,100),VTCOM(100),VNCOM(100)           INT01710
     +    ,XS(100),YS(100),SOFF(100),COFF(100),XPOFF(100),YPOFF(100)    INT01720
     +    ,Y(100),SIG(3,100),DLS(100),DLSC(100),A(100,100),B(100,100)   INT01730
     +    ,XSTGR(100),YSTGR(100),VUT(3),VLT(3),VUN(3),VLN(3),DBPP(100)  INT01740
     +    ,CPI(100),XOS(100),YOS(100),DBPPC(100)                        INT01750
      REAL NUM1 , NUM2                                                  INT01760
      LOGICAL OFF,SINGLE                                                INT01770
      OFF = .FALSE.                                                     INT01780
      PI = 3.141592                                                     INT01790
      CM = 0.0                                                          INT01800
      N = N1 - 1                                                        INT01810
      IF (ICYCLE .EQ. 1) THEN                                           INT01820
      IF (IRST .EQ. 0) THEN                                             INT01830
      DO 10 I=1,N1                                                      INT01840
      DLS(I) = 0.0                                                      INT01850
      VN (I) = 0.0                                                      INT01860
      DBPP(I)= 0.0                                                      INT01870
10    CONTINUE                                                          INT01880
      ELSE                                                              INT01890
      DO 5 I = 1 , N1                                                   INT01900
      XOS(I) = XO(I)                                                    INT01910
      YOS(I) = YO(I)                                                    INT01920
5     CONTINUE                                                          INT01930
      READ (3) NT,(XS(I),YS(I),DLSC(I),VNC(I),DBPPC(I),I=1,NT)          INT01940
      XMIN = XS(1)                                                      INT01950
      DO 15 I = 2 , NT                                                  INT01960
      IF (XS(I) .GT. XMIN) GOTO 15                                      INT01970
      XMIN = XS(I)                                                      INT01980
      IMIN = I                                                          INT01990
15    CONTINUE                                                          INT02000
      DO 17 I = 1 , NT                                                  INT02010
      IF (I .LT. IMIN) GOTO 16                                          INT02020
      XS(I) = XS(I) - XMIN                                              INT02030
      GOTO 17                                                           INT02040
16    XS(I) = XMIN - XS(I)                                              INT02050
17    CONTINUE                                                          INT02060
C                                                                       INT02070
      XMIN = XOS(I)                                                     INT02080
      DO 20 I = 2 , N1                                                  INT02090
      IF (XOS(I) .GT. XMIN) GOTO 20                                     INT02100
      XMIN = XOS(I)                                                     INT02110
      IMIN = I                                                          INT02120
20    CONTINUE                                                          INT02130
      DO 22 I = 1 , N1                                                  INT02140
      IF (I .LT. IMIN) GOTO 21                                          INT02150
      XOS(I) = XOS(I) - XMIN                                            INT02160
```

```
      GOTO 22                                                          INT0217
21    XOS(I) = XMIN - XOS(I)                                           INT0218
22    CONTINUE                                                         INT0219
C                                                                      INT0220
      CALL DIFF3(NT,XS,DLSC,D1,D2,D3,0)                                INT0221
      CALL INTRP3(NT,XS,DLSC,D1,D2,D3,N1,XOS,DLS)                      INT0222
      CALL AMEAN (1,N1,XOS,DLS,1)                                      INT0223
      CALL DIFF3 (NT,XS,VNC,D1,D2,D3,0)                                INT0224
      CALL INTRP3(NT,XS,VNC,D1,D2,D3,N1,XOS,VN)                        INT0225
      CALL AMEAN (1,N1,XOS,VN,1)                                       INT0226
      CALL DIFF3 (NT,XS,DBPPC,D1,D2,D3,0)                              INT0227
      CALL INTRP3(NT,XS,DBPPC,D1,D2,D3,N1,XOS,DBPP)                    INT0228
      CALL AMEAN (1,N1,XOS,DBPP,1)                                     INT0229
      END IF                                                           INT0230
      END IF                                                           INT0231
      DO 30 I=1,N1                                                     INT0232
      XP(I) = XSTGR(I)                                                 INT0233
      YP(I) = YSTGR(I)                                                 INT0234
30    CONTINUE                                                         INT0235
C  CALCULATE GEOMETRIC QUANTITIES                                      INT0236
      DO 100 J=1,N                                                     INT0237
      VNC(J) = 0.5 * (VN(J) + VN(J+1))                                 INT0238
      X(J)= .5*(XP(J)+XP(J+1))                                         INT0239
      Y(J)= .5*(YP(J)+YP(J+1))                                         INT0240
      D(J)= SQRT((XP(J+1)-XP(J))**2 + (YP(J+1)-YP(J))**2)              INT0241
      C(J)= (XP(J+1)-XP(J))/D(J)                                       INT0242
      S(J)= (YP(J+1)-YP(J))/D(J)                                       INT0243
100   CONTINUE                                                         INT0244
C                                                                      INT0245
      IF ( INLET .NE. 0 .AND. .NOT. SINGLE) THEN                       INT0246
      SUM = D(1)                                                       INT0247
      DO 35 J = 2, N                                                   INT0248
      SUM = SUM + D(J)                                                 INT0249
35    CONTINUE                                                         INT0250
      Q = 2.0 * PI * SUM / SP                                          INT0251
      ELSE                                                             INT0252
      Q = 0.0                                                          INT0253
      END IF                                                           INT0254
C                                                                      INT0255
C  CALCULATE NORMAL AND TANGENTIAL MATRICES                            INT0256
102   CONTINUE                                                         INT0257
      IF (SINGLE) THEN                                                 INT0258
      IF ( .NOT. OFF) THEN                                             INT0259
      DO 120 I=1,N                                                     INT0260
      DO 110 J=1,N                                                     INT0261
      IF (J .EQ. I) GO TO 105                                          INT0262
      XX= (X(I)-X(J))*C(J) + (Y(I)-Y(J))*S(J)                          INT0263
      YY=-(X(I)-X(J))*S(J) + (Y(I)-Y(J))*C(J)                          INT0264
      UU= LOG(((XX+.5*D(J))**2+YY**2)/((XX-.5*D(J))**2+YY**2))         INT0265
      VV= 2.*ATAN2(YY*D(J), XX**2+YY**2-(.5*D(J))**2)                  INT0266
      SS= S(I)*C(J) - C(I)*S(J)                                        INT0267
      CC= C(I)*C(J) + S(I)*S(J)                                        INT0268
      A(I,J)= -UU*SS + VV*CC                                           INT0269
      B(I,J)=  UU*CC + VV*SS                                           INT0270
      GO TO 110                                                        INT0271
105   A(I,J) = 6.2831853                                               INT0272
```

```
      B(I,J) = 0.0                                                          INT02730
110   CONTINUE                                                              INT02740
120   CONTINUE                                                              INT02750
      ELSE                                                                  INT02760
      DO 140 I=1,N                                                          INT02770
      DO 130 J=1,N                                                          INT02780
      XX= (XOFF(I)-X(J))*C(J) + (YOFF(I)-Y(J))*S(J)                         INT02790
      YY=-(XOFF(I)-X(J))*S(J) + (YOFF(I)-Y(J))*C(J)                         INT02800
      UU= LOG(((XX+.5*D(J))**2+YY**2)/((XX-.5*D(J))**2+YY**2))              INT02810
      VV= 2.*ATAN2(YY*D(J), XX**2+YY**2-(.5*D(J))**2)                       INT02820
      SS= SOFF(I)*C(J) - COFF(I)*S(J)                                       INT02830
      CC= COFF(I)*C(J) + SOFF(I)*S(J)                                       INT02840
      AOFF(I,J)= -UU*SS + VV*CC                                             INT02850
      BOFF(I,J)=  UU*CC + VV*SS                                             INT02860
130   CONTINUE                                                              INT02870
140   CONTINUE                                                              INT02880
      END IF                                                                INT02890
      ELSE                                                                  INT02900
      IF ( .NOT. OFF) THEN                                                  INT02910
      DO 50 I=1,N                                                           INT02920
      DO 40 J=1,N                                                           INT02930
      IF (J .EQ. I) GO TO 45                                                INT02940
      XX= (X(I)-X(J))*C(J) + (Y(I)-Y(J))*S(J)                               INT02950
      YY=-(X(I)-X(J))*S(J) + (Y(I)-Y(J))*C(J)                               INT02960
      DX1 = PI *(X(I)-XP(J)) / SP                                           INT02970
      DY1 = PI *(Y(I)-YP(J)) / SP                                           INT02980
      DX2 = PI *(X(I)-XP(J+1)) / SP                                         INT02990
      DY2 = PI *(Y(I)-YP(J+1)) / SP                                         INT03000
      R1SQ = (COSH(DX1))**2 - (COS(DY1))**2                                 INT03010
      R2SQ = (COSH(DX2))**2 - (COS(DY2))**2                                 INT03020
      UU   = LOG(R1SQ/R2SQ)                                                 INT03030
      NUM1 = DX1 * COSH(DX1) * SIN(DY1) -                                   INT03040
     +       DY1 * SINH(DX1) * COS(DY1)                                     INT03050
      DNUM1= DX1 * SINH(DX1) * COS(DY1) +                                   INT03060
     +       DY1 * COSH(DX1) * SIN(DY1)                                     INT03070
      NUM2 = DX2 * COSH(DX2) * SIN(DY2) -                                   INT03080
     +       DY2 * SINH(DX2) * COS(DY2)                                     INT03090
      DNUM2= DX2 * SINH(DX2) * COS(DY2) +                                   INT03100
     +       DY2 * COSH(DX2) * SIN(DY2)                                     INT03110
      EXV  = 2.0 * ATAN2(NUM2,DNUM2) - 2.0 * ATAN2(NUM1,DNUM1)              INT03120
      VV= 2.*ATAN2(YY*D(J), XX**2+YY**2-(.5*D(J))**2)                       INT03130
      VV = VV + EXV                                                         INT03140
      SS= S(I)*C(J) - C(I)*S(J)                                             INT03150
      CC= C(I)*C(J) + S(I)*S(J)                                             INT03160
      A(I,J)= -UU*SS + VV*CC                                                INT03170
      B(I,J)=  UU*CC + VV*SS                                                INT03180
      GO TO 40                                                              INT03190
45    A(I,J) = 6.2831853                                                    INT03200
      B(I,J) = 0.0                                                          INT03210
40    CONTINUE                                                              INT03220
50    CONTINUE                                                              INT03230
      ELSE                                                                  INT03240
      DO 70 I=1,N                                                           INT03250
      DO 60 J=1,N                                                           INT03260
      XX= (XOFF(I)-X(J))*C(J) + (YOFF(I)-Y(J))*S(J)                         INT03270
      YY=-(XOFF(I)-X(J))*S(J) + (YOFF(I)-Y(J))*C(J)                         INT03280
```

```
      DX1 = PI *(XOFF(I)-XP(J)) / SP                                    INT03290
      DY1 = PI *(YOFF(I)-YP(J)) / SP                                    INT03300
      DX2 = PI *(XOFF(I)-XP(J+1)) / SP                                  INT03310
      DY2 = PI *(YOFF(I)-YP(J+1)) / SP                                  INT03320
      R1SQ = (COSH(DX1))**2 - (COS(DY1))**2                             INT03330
      R2SQ = (COSH(DX2))**2 - (COS(DY2))**2                             INT03340
      UU   = LOG(R1SQ/R2SQ)                                             INT03350
      NUM1 = DX1 * COSH(DX1) * SIN(DY1) -                               INT03360
     +       DY1 * SINH(DX1) * COS(DY1)                                 INT03370
      DNUM1= DX1 * SINH(DX1) * COS(DY1) +                               INT03380
     +       DY1 * COSH(DX1) * SIN(DY1)                                 INT03390
      NUM2 = DX2 * COSH(DX2) * SIN(DY2) -                               INT03400
     +       DY2 * SINH(DX2) * COS(DY2)                                 INT03410
      DNUM2= DX2 * SINH(DX2) * COS(DY2) +                               INT03420
     +       DY2 * COSH(DX2) * SIN(DY2)                                 INT03430
      EXV  = 2.0 * ATAN2(NUM2,DNUM2) - 2.0 * ATAN2(NUM1,DNUM1)          INT03440
      VV= 2.*ATAN2(YY*D(J), XX**2+YY**2-(.5*D(J))**2)                   INT03450
      VV = VV + EXV                                                     INT03460
      SS= SOFF(I)*C(J) - COFF(I)*S(J)                                   INT03470
      CC= COFF(I)*C(J) + SOFF(I)*S(J)                                   INT03480
      AOFF(I,J)= -UU*SS + VV*CC                                         INT03490
      BOFF(I,J)=  UU*CC + VV*SS                                         INT03500
60    CONTINUE                                                          INT03510
70    CONTINUE                                                          INT03520
      END IF                                                            INT03530
      END IF                                                            INT03540
C  NORMAL AND TANGENTIAL COMPONENTS OF FUNDAMENTAL SOLUTIONS            INT03550
C                                                                       INT03560
      DO 160 I=1,N                                                      INT03570
      SUMR= 0.                                                          INT03580
      SUMT= 0.                                                          INT03590
      IF (  .NOT. OFF) THEN                                             INT03600
      R(1,I)= S(I)+VNC(I)/COS(ALPHA)                                    INT03610
      T(1,I)= C(I)                                                      INT03620
      R(2,I)= -C(I)                                                     INT03630
      T(2,I)= S(I)                                                      INT03640
      DO 145 J=1,N                                                      INT03650
      SUMR = SUMR + B(I,J)                                              INT03660
      SUMT = SUMT + A(I,J)                                              INT03670
145   CONTINUE                                                          INT03680
      ELSE                                                              INT03690
      R(1,I) = SOFF(I)                                                  INT03700
      T(1,I) = COFF(I)                                                  INT03710
      R(2,I) =-COFF(I)                                                  INT03720
      T(2,I) = SOFF(I)                                                  INT03730
      DO 150 J=1,N                                                      INT03740
      SUMR= SUMR + BOFF(I,J)                                            INT03750
      SUMT= SUMT + AOFF(I,J)                                            INT03760
150   CONTINUE                                                          INT03770
      END IF                                                            INT03780
      R(3,I)= SUMR                                                      INT03790
      T(3,I)= SUMT                                                      INT03800
160   CONTINUE                                                          INT03810
C                                                                       INT03820
      IF ( OFF ) GO TO 275                                              INT03830
C     DECOMPOSITION OF MATRIX A                                         INT03840
```

```
      DO 230 I=1,N-1                                                    INT03850
      DO 220 K=I+1,N                                                    INT03860
      A(K,I)= A(K,I)/A(I,I)                                             INT03870
      DO 210 J=I+1,N                                                    INT03880
      A(K,J)= A(K,J)- A(K,I)*A(I,J)                                     INT03890
210   CONTINUE                                                          INT03900
220   CONTINUE                                                          INT03910
230   CONTINUE                                                          INT03920
C  OPERATE ON FUNDAMENTAL RIGHT SIDES WITH LOWER TRIANGULAR             INT03930
      DO 270 K=1,3                                                      INT03940
      DO 260 J=1,N-1                                                    INT03950
      DO 250 I=J+1,N                                                    INT03960
      R(K,I)= R(K,I) - A(I,J)*R(K,J)                                    INT03970
250   CONTINUE                                                          INT03980
260   CONTINUE                                                          INT03990
270   CONTINUE                                                          INT04000
C  BACK SOLUTION                                                        INT04010
      DO 300 K=1,3                                                      INT04020
      DO 290 I=N,1,-1                                                   INT04030
      SUM= 0.                                                           INT04040
      DO 280 J=N,I+1,-1                                                 INT04050
      SUM= SUM + A(I,J)*SIG(K,J)                                        INT04060
280   CONTINUE                                                          INT04070
      SIG(K,I)= (R(K,I)-SUM)/A(I,I)                                     INT04080
290   CONTINUE                                                          INT04090
300   CONTINUE                                                          INT04100
      OFF = .TRUE.                                                      INT04110
C                                                                       INT04120
C    ADD DIS-PLACE VERTICALLY TO THE BODY TO GENERATE                   INT04130
C    DISPLACEMENT SURFACE                                               INT04140
C                                                                       INT04150
      DO 305 I = 2 , N                                                  INT04160
      SOFF(I) = (S(I)*D(I-1)+S(I-1)*D(I))/(D(I-1)+D(I))                 INT04170
      COFF(I) = (C(I)*D(I-1)+C(I-1)*D(I))/(D(I-1)+D(I))                 INT04180
305   CONTINUE                                                          INT04190
      SOFF(1) = 2.0*S(1) - SOFF(2)                                      INT04200
      SOFF(N1)= 2.0*S(N) - SOFF(N)                                      INT04210
      COFF(1) = 2.0*C(1) - COFF(2)                                      INT04220
      COFF(N1)= 2.0*C(N) - COFF(N)                                      INT04230
      DO 306 I = 1 , N1                                                 INT04240
      XPOFF(I) = XP(I) - SOFF(I) * DLS(I)                               INT04250
      YPOFF(I) = YP(I) + COFF(I) * DLS(I)                               INT04260
306   CONTINUE                                                          INT04270
      DO 307 I = 1 , N                                                  INT04280
      XOFF(I) = 0.5 * (XPOFF(I) + XPOFF(I+1))                           INT04290
      YOFF(I) = 0.5 * (YPOFF(I) + YPOFF(I+1))                           INT04300
      DOFF = SQRT((XPOFF(I+1)-XPOFF(I))**2 +                            INT04310
     +            (YPOFF(I+1)-YPOFF(I))**2 )                            INT04320
      COFF(I) = (XPOFF(I+1)-XPOFF(I))/DOFF                              INT04330
      SOFF(I) = (YPOFF(I+1)-YPOFF(I))/DOFF                              INT04340
307   CONTINUE                                                          INT04350
      GO TO 102                                                         INT04360
C  CALCULATION OF SURFACE VELOCITIES FOR THE FUNDAMENTAL SOLUTIONS      INT04370
C                                                                       INT04380
275   DO 330 K=1,3                                                      INT04390
      DO 320 I=1,N                                                      INT04400
```

```
      SUMT= T(K,I)                                                      INT04410
      SUMN=-R(K,I)                                                      INT04420
      DO 310 J=1,N                                                      INT04430
      SUMT= SUMT + BOFF(I,J)*SIG(K,J)                                   INT04440
      SUMN= SUMN + AOFF(I,J)*SIG(K,J)                                   INT04450
310   CONTINUE                                                          INT04460
      VTAN(K,I)= SUMT                                                   INT04470
      VNOR(K,I)= SUMN                                                   INT04480
320   CONTINUE                                                          INT04490
C                                                                       INT04500
      DOFF1 = SQRT((XPOFF(2)-XPOFF(1))**2+(YPOFF(2)-YPOFF(1))**2)       INT04510
      DOFF2 = SQRT((XPOFF(3)-XPOFF(2))**2+(YPOFF(3)-YPOFF(2))**2)       INT04520
      DOFFN = SQRT((XPOFF(N+1)-XPOFF(N))**2+(YPOFF(N+1)-YPOFF(N))       INT04530
     +              **2)                                                INT04540
      DOFFN1= SQRT((XPOFF(N)-XPOFF(N-1))**2+(YPOFF(N)-YPOFF(N-1))       INT04550
     +              **2)                                                INT04560
      VUT(K) = VTAN(K,N) + DOFFN * (VTAN(K,N)-VTAN(K,N-1))/             INT04570
     +                             (DOFFN+DOFFN1)                       INT04580
      VUN(K) = VNOR(K,N) + DOFFN * (VNOR(K,N)-VNOR(K,N-1))/             INT04590
     +                             (DOFFN+DOFFN1)                       INT04600
      VLT(K) = VTAN(K,1) + DOFF1 * (VTAN(K,1)-VTAN(K,2))/               INT04610
     +                             (DOFF1+DOFF2)                        INT04620
      VLN(K) = VNOR(K,1) + DOFF1 * (VNOR(K,1)-VNOR(K,2))/               INT04630
     +                             (DOFF1+DOFF2)                        INT04640
330   CONTINUE                                                          INT04650
C  OUTPUT FUNDAMENTAL SOLUTIONS                                         INT04660
      IF (ICYCLE .EQ. 1 .OR. ICYCLE .GE. ICYTL-1 .OR. IP .GE. 0)        INT04670
     + WRITE(6,335) TITLE                                               INT04680
335   FORMAT(1H1,///20A4//)                                             INT04690
C     DO 360 K=1,3                                                      INT04700
C     WRITE (6,340) K                                                   INT04710
C340   FORMAT(////,1H ,'FUNDAMENTAL SOLUTION NUMBER ',I2////)           INT04720
C      WRITE(6,345)                                                     INT04730
345   FORMAT(3X,'I',8X,'X',11X,'Y',10X,'VT',10X,'VN',8X,'SIG' ///)      INT04740
C      WRITE(6,375) 1,  XP(1) , YP(1), VLT(K),VLN(K)                    INT04750
C      DO 350 I=1,N                                                     INT04760
C      WRITE(6,375) I , X(I), Y(I), VTAN(K,I),VNOR(K,I),SIG(K,I)        INT04770
C350   CONTINUE                                                         INT04780
C      WRITE(6,375) N1 , XP(N1),YP(N1),VUT(K),VUN(K)                    INT04790
C360   CONTINUE                                                         INT04800
C  COMBINED FLOW AT ANGLE OF ATTACK                                     INT04810
C                                                                       INT04820
      IF ( INLET .NE. 0) THEN                                           INT04830
      YYY = ((VUT(3)+VLT(3))*TAN(ALPHAI)+(VUT(1)+VLT(1))*Q)/            INT04840
     +      ((VUT(3)+VLT(3))-(VUT(2)+VLT(2))*Q)                         INT04850
      XXX = -((VUT(1)+VLT(1))+(VUT(2)+VLT(2))*TAN(ALPHAI))/             INT04860
     +       ((VUT(3)+VLT(3))-(VUT(2)+VLT(2))*Q)                        INT04870
      ALPHA = ACOS(1.0/SQRT(1.0+YYY**2))                                INT04880
      COSAL = COS(ALPHA)                                                INT04890
      SINAL = SIN(ALPHA)                                                INT04900
      W = XXX/SQRT(1.0+YYY**2)                                          INT04910
      ELSE                                                              INT04920
      COSAL = COS(ALPHA)                                                INT04930
      SINAL = SIN(ALPHA)                                                INT04940
      W=-((VLT(1)+VUT(1))*COSAL+(VLT(2)+VUT(2))*SINAL)/                 INT04950
     +       (VLT(3)+VUT(3))                                            INT04960
```

```
      END IF                                                            INT04970
C  FORCE COEFFICIENT CALCULATION                                        INT04980
      SUM1= 0.                                                          INT04990
      SUMX= 0.                                                          INT05000
      SUMY= 0.                                                          INT05010
      DO 390 I=1,N                                                      INT05020
      SUM1= SUM1+ D(I)                                                  INT05030
      SUMX= SUMX- VCOM(I)**2*S(I)*D(I)                                  INT05040
      SUMY= SUMY+ VCOM(I)**2*C(I)*D(I)                                  INT05050
390   CONTINUE                                                          INT05060
C  FIND MAN. CHORD LENGTH                                               INT05070
      XOMIN = XO(1)                                                     INT05080
      DO 395 I = 2 , N1                                                 INT05090
      IF ( XO(I) .GT. XOMIN) GOTO 395                                   INT05100
      XOMIN = XO(I)                                                     INT05110
395   CONTINUE                                                          INT05120
      CHORD = XO(N1) - XOMIN                                            INT05130
      CL1= SUM1*25.13274*W/CHORD                                        INT05140
      CL2= (SUMY*COSAL-SUMX*SINAL)/CHORD                                INT05150
      CD = (SUMX*COSAL+SUMY*SINAL)/CHORD                                INT05160
C                                                                       INT05170
C  CALCULATING PARAMETERS FOR INLET VELOCITY AS MODULUS OF NOMORIZED VELINT05180
C                                                                       INT05190
      IF (.NOT. SINGLE) THEN                                            INT05200
      NUM1 = SIN(ALPHA)+CL1*CHORD/(4.0*SP)                              INT05210
      ALPHID = ATAN2(NUM1,COS(ALPHA))                                   INT05220
      NUM1 = SIN(ALPHA)-CL1*CHORD/(4.0*SP)                              INT05230
      ALPHED = ATAN2(NUM1,COS(ALPHA))                                   INT05240
      NUM1 = CL1*CHORD/(2.0*SP)*COS(ALPHA)                              INT05250
      DNUM1= 1.0-(CL1*CHORD/(4.0*SP))**2                                INT05260
      DALPHA = ATAN2(NUM1,DNUM1)                                        INT05270
      UOUI = (TAN(ALPHID)-TAN(ALPHED))*(2.0*SP/CHORD*COS(ALPHID))/CL1   INT05280
      CLI = CL1 * UOUI**2                                               INT05290
      UIOU = 1.0/UOUI                                                   INT05300
      VEXIT = COS(ALPHA)/COS(ALPHED)                                    INT05310
      ELSE                                                              INT05320
      ALPHID = ALPHA                                                    INT05330
      ALPHED = ALPHA                                                    INT05340
      DALPHA = 0.0                                                      INT05350
      UOUI = 1.0                                                        INT05360
      UIOU = 1.0                                                        INT05370
      CLI  = CL1                                                        INT05380
      VEXIT = 1.0                                                       INT05390
      END IF                                                            INT05400
      FAC = 180.0/PI                                                    INT05410
      ALPHID = ALPHID * FAC                                             INT05420
      ALPHED = ALPHED * FAC                                             INT05430
      DALPHA = DALPHA * FAC                                             INT05440
      ALPHAD = ALPHA  * FAC                                             INT05450
      IF (ICYCLE .EQ. 1 .OR. ICYCLE .GE. ICYTL-1 .OR. IP .GE. 0) THEN   INT05460
      WRITE(6,370) ALPHAD ,ALPHID,ALPHED,DALPHA,UIOU,VEXIT              INT05470
370   FORMAT (////,1H , 'COMBINED FLOW AT AVERAGE ANGLE OF ATTACK = ',  INT05480
     +          F8.3, 4X,'DEGREES', /,1H ,17X,'INLET ANGLE OF ',       INT05490
     +         'ATTACK   = ',F8.3,4X,'DEGREES',/,1H ,                   INT05500
     +          17X,'EXIT ANGLE =  ',F8.3,4X,'DEGREES',/,1H ,17X,      INT05510
     +          'TURNNING ANGLE = ',F8.3,4X,'DEGREES',/,1H ,17X,       INT05520
```

```
     +          'INLET VEL = ',F10.6,3X,'EXIT VEL = ',F10.6,//)          INT05530
      WRITE(6,365)                                                       INT05540
365   FORMAT(3X,'I',8X,'XO',10X,'YO',10X,'X',11X,'Y',10X,'VT',           INT05550
     +           10X,'VN',11X,'V',10X,'CP',9X,'CPI'///)                  INT05560
      END IF                                                             INT05570
      DO 380 I=1,N                                                       INT05580
      VTCOM(I)= VTAN(1,I)*COSAL+VTAN(2,I)*SINAL+W*VTAN(3,I)              INT05590
      VNCOM(I)= VNOR(1,I)*COSAL+VNOR(2,I)*SINAL+W*VNOR(3,I)              INT05600
      VCOM(I)= SQRT(VTCOM(I)**2 + VNCOM(I)**2)                           INT05610
      IF (VTCOM(I) .LT. 0.0) VCOM(I) = -VCOM(I)                          INT05620
      CP(I) = 1.0 - VCOM(I) ** 2                                         INT05630
      CPI(I)= 1.0 - (VCOM(I)*UOUI)**2                                    INT05640
      SIGCOM(I) = SIG(1,I)*COSAL+SIG(2,I)*SINAL+W*SIG(3,I)               INT05650
      XP(I) = 0.5 * (XO(I)+XO(I+1))                                      INT05660
      YP(I) = 0.5 * (YO(I)+YO(I+1))                                      INT05670
380   CONTINUE                                                           INT05680
      IF (ICYCLE .EQ. 1 .OR. ICYCLE .GE. ICYTL-1 .OR. IP .GE. 0) THEN    INT05690
      WRITE (1,374)  ( XO(I),YO(I),XP(I),YP(I),CP(I),CPI(I) ,I=1,N)      INT05700
      WRITE (6,375)  ( I, XO(I), YO(I), XP(I), YP(I), VTCOM(I),          INT05710
     +                 VNCOM(I),VCOM(I),CP(I), CPI(I) ,I=1,N)          INT05720
374   FORMAT(6F10.4)                                                     INT05730
375   FORMAT(1X, I3, 9F12.4)                                             INT05740
C     WRITE (2) I,XO(I),YO(I),XSTGR(I),YSTGR(I),DLS(I),X(I),Y(I),VCOM(I)INT05750
      WRITE(6,385) N+1,XO(N+1),YO(N+1)                                   INT05760
385   FORMAT(1X,I3,2F12.4)                                               INT05770
      WRITE(6,400) CHORD, CL1 ,CLI                                       INT05780
400   FORMAT(///3X,'CHORD = ',F10.5,4X,'CL(AVG) = ',F10.5,4X,            INT05790
     +          'CL(INLET) = ',F10.5)                                    INT05800
      END IF                                                             INT05810
420   FORMAT(/3X,1HI,6X,2HSO,10X,2HSC,9X,3H VN,9X,3HVNC,9X,3HDLS,8X,     INT05820
     +           4HDLSC)                                                 INT05830
430   FORMAT(I5,6E12.4)                                                  INT05840
      RETURN                                                             INT05850
      END                                                                INT05860
C                                                                        INT05870
C  ---------------------------------------------                         INT05880
C          DATA SET KCBCAMEAN  AT LEVEL 001 AS OF 08/24/84               INT05890
C          DATA SET KCBCAMEAN  AT LEVEL 003 AS OF 04/05/84               INT05900
      SUBROUTINE AMEAN(NS,ND,X,Y,IT)                                     INT05910
C                                                                        INT05920
C   SMOOTH DATA USING 3-PTS WEIGHTING FORMULA                            INT05930
C   NS   :  STARTING PINT OF THE DATA TO BE SMOOTHED                     INT05940
C   ND   :  END PINT OF THE DATA TO BE SMOOTHED                          INT05950
C   X, Y :  INDEPENDENT + DEPENDENT VARAIBLES OF THE DATA                INT05960
C           TO BE SMOOTHED                                               INT05970
C   IT   :  CYCLES OF DATA SMOOTHING                                     INT05980
C                                                                        INT05990
      DIMENSION X(101),Y(101)                                            INT06000
C                                                                        INT06010
C ---------------------------------------------------------------        INT06020
C                                                                        INT06030
      NM    = ND -NS                                                     INT06040
      IF(NM .LT.2 .OR.  IT .LT.1) RETURN                                 INT06050
C                                                                        INT06060
      NDM1  = ND - 1                                                     INT06070
      NSP1  = NS + 1                                                     INT06080
```

```
      DO 20 K=1,IT                                                    INT06090
      DL1   = X(NSP1) - X(NS)                                         INT06100
      Y1    = Y(NS)                                                   INT06110
      DO 10 I=NSP1,NDM1                                               INT06120
      DL2   = X(I + 1) -X(I)                                          INT06130
      Y2    = Y(I)                                                    INT06140
      YM    = (DL2 * Y1 + DL1 * Y(I+1))/(DL1 + DL2)                   INT06150
      Y(I)  = 0.5 * (Y2 + YM)                                         INT06160
      DL1   = DL2                                                     INT06170
      Y1    = Y2                                                      INT06180
10    CONTINUE                                                        INT06190
20    CONTINUE                                                        INT06200
C                                                                     INT06210
      RETURN                                                          INT06220
      END                                                             INT06230
C          DATA SET KCBCBLGRID AT LEVEL 001 AS OF 08/24/84            INT06240
C          DATA SET KCBCBLGRID AT LEVEL 001 AS OF 08/24/84            INT06250
C          DATA SET KCBCBLGRID AT LEVEL 004 AS OF 04/05/84            INT06260
      SUBROUTINE BLGRID(N,X,T,D1)                                     INT06270
C                                                                     INT06280
C   GENERATE B. L. X-WISE GRID USING MODIFIED COSINE DISTRIBUTION     INT06290
C                                                                     INT06300
      DIMENSION X(101),T(101),D1(101)                                 INT06310
      DATA CRAD/57.2957795/, BPI/3.14159265/                          INT06320
C                                                                     INT06330
C   ------------------------------------------------------------      INT06340
C                                                                     INT06350
      NN     = 2 * N - 1                                              INT06360
      EN     = FLOAT((NN-1)/2)                                        INT06370
      THO    = 10./CRAD                                               INT06380
      CTO1   = 1. + COS(THO)                                          INT06390
      DTH    = (BPI - THO) / EN                                       INT06400
      FI     = FLOAT(N - 2)                                           INT06410
      DO 10 I=N,NN                                                    INT06420
      FI     = 1.0 + FI                                               INT06430
      II     = I - N + 1                                              INT06440
      XII    = THO + FI * DTH                                         INT06450
      X(II)  = (1.0 + COS(XII))/CTO1                                  INT06460
10    CONTINUE                                                        INT06470
      X1     = X(1)                                                   INT06480
      XN     = X(N)                                                   INT06490
      CH     = XN -X1                                                 INT06500
      FN1    = FLOAT(N-1)                                             INT06510
      N10     = N/10                                                  INT06520
      DO 20 I=1,N                                                     INT06530
      T(I)   = FLOAT(I-1)/FN1                                         INT06540
      X(I)   = (X(I)-X1)/CH                                           INT06550
20    CONTINUE                                                        INT06560
C     CALL SMFIT(N10,N,T,X,D1,N10)                                    INT06570
C     IF(X(2).LT.0.35 * X(3)) X(2) = 0.35 * X(3)                      INT06580
C     CALL SMFIT(1,N,T,X,D1,2)                                        INT06590
      CALL AMEAN(1,N,T,X,N10)                                         INT06600
C                                                                     INT06610
      RETURN                                                          INT06620
      END                                                             INT06630
C          DATA SET KCBCBL2D   AT LEVEL 001 AS OF 08/24/84            INT06640
```

```
C           DATA SET KCBCBL2D   AT LEVEL 001 AS OF 08/24/84                INT06650
C           DATA SET KCBCBL2D   AT LEVEL 012 AS OF 04/06/84                INT06660
      SUBROUTINE BL2D ( ITR,ISWPT,SURFID)                                   INT06670
C   PROGRAM CALCULATES VISCOUS/INVISCID INTERACTION USING HILBERT           INT06680
C   INTEGRAL.                                                               INT06690
C                                                                           INT06700
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                        INT06710
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)            INT06720
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)                 INT06730
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI          INT06740
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100),ALFAS(100),            INT06750
     +              FFS(100),RTS(100),IEDY,NXSPT                            INT06760
      COMMON /SMRY/ VW(100),ITP(100),ISL(100),DLS(100),CF(100),             INT06770
     +              THT(100),NPSTR(100)                                     INT06780
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH                 INT06790
      COMMON /BONV/ ITMAX,EPSL,EPST,CONV                                    INT06800
      COMMON /SAVE/ FS(101),US(101),VS(101),WS(101),BS(101)                 INT06810
      COMMON /BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),           INT06820
     +            XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)            INT06830
      COMMON /1SURF/ ISF                                                    INT06840
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                     INT06850
      DIMENSION SURFID(4)                                                   INT06860
C                                                                           INT06870
C     ------------------------------------------------------------          INT06880
C                                                                           INT06890
C     GENERATE B. L. GRIDS + SET INITIAL CONDITIONS                         INT06900
C                                                                           INT06910
      DO 5 I = 1 , NXT                                                      INT06920
      ALFAS(I) = 0.0                                                        INT06930
      FFS(I)   = 1.0                                                        INT06940
      RTS(I)   = 1.0                                                        INT06950
5     CONTINUE                                                              INT06960
      CALL INPUT(ITR,ISWPT,SURFID)                                          INT06970
C                                                                           INT06980
C     CALCULATE HILBERT COEFFS., C(I,J)                                     INT06990
C                                                                           INT07000
      CALL CALCIJ(NXT,0)                                                    INT07010
C                                                                           INT07020
C   LOOP OF CALCULATIONS                                                    INT07030
C                                                                           INT07040
      NSS   = NS                                                            INT07050
      NXSPT = NXT + 1                                                       INT07060
C     IF ( ICYCLE .EQ. 1 ) NS = NXT + 1                                     INT07070
10    NX    = NX + 1                                                        INT07080
      CEL   = 0.5 * (X(NX) + X(NX-1)) /(X(NX) -X(NX-1))                     INT07090
      CELH  = 0.5 * CEL                                                     INT07100
20    IT    = 0                                                             INT07110
      RX    = UE(NX)*X(NX)*RL                                               INT07120
      SQRX  = SQRT(RX)                                                      INT07130
30    IT    = IT + 1                                                        INT07140
      IF(IT .LE. ITMAX) GO TO 40                                            INT07150
      NXM1   = NX-1                                                         INT07160
      CALL HEADER( TITLE,SURFID,ISTRP )                                     INT07170
      WRITE(6, 170 ) (M,X(M),CF(M),DLS(M),UE(M),P2  (M),THT(M),             INT07180
     +              D(M),ALFAS(M),  ITP(M),NPSTR(M),M=1,NXM1)               INT07190
      WRITE(6, 160 ) NX                                                     INT07200
```

```
      STOP                                                              INT07210
40    CONTINUE                                                          INT07220
      IF(NX .GT. NTR) CALL EDDY                                         INT07230
      CALL COEFTR                                                       INT07240
      CALL SOLV3                                                        INT07250
      IF(V(1,2).GT.0.0) GOTO 60                                         INT07260
C                                                                       INT07270
C     EXTRAPOLATE CALCULATED D FOR TURBULENT SEPARATION OR LAMINAR      INT07280
C     SEPARATION FOR LAMINAR FLOW CALCULATION ONLY                      INT07290
C                                                                       INT07300
      CALL EXTRAP(NX,NXT,X,D)                                           INT07310
      NXM1 = NX - 1                                                     INT07320
      CALL HEADER( TITLE,SURFID,ISTRP )                                 INT07330
      WRITE(6, 170 ) (M,X(M),CF(M),DLS(M),UE(M),P2(M),THT(M),           INT07340
     +                D(M),ALFAS(M),ITP(M),NPSTR(M),M=1,NXM1)          INT07350
      WRITE(6,180)                                                      INT07360
      WRITE(6,190) (M,X(M),D(M),M=NX,NXT)                               INT07370
      GOTO 130                                                          INT07380
60    IF(NX .GT. NTR) GO TO 70                                          INT07390
      IF(ABS(DELV(1)) .GT. EPSL) GO TO 30                               INT07400
      GO TO 80                                                          INT07410
70    CONTINUE                                                          INT07420
      IF(ABS(DELV(1)/V(1,2)) .GT. EPST) GO TO 30                        INT07430
80    CONTINUE                                                          INT07440
C                                                                       INT07450
C     CHECK FOR GROWTH                                                  INT07460
C                                                                       INT07470
      IF(NP .GE. NPT) GO TO 90                                          INT07480
      IF(ABS(V(NP,2)) .LT. 0.0005 .AND. ABS(1.0-U(NP-2,2))              INT07490
     +          .LT.0.0035) GOTO 90                                     INT07500
      CALL FILLUP(1)                                                    INT07510
      IT = 1                                                            INT07520
      GO TO 30                                                          INT07530
90    CONTINUE                                                          INT07540
C                                                                       INT07550
      CALL FILLUP(2)                                                    INT07560
      CALL OUTPUT(1)                                                    INT07570
      IF(ITR.EQ.0 .OR. NX.GE.NTR) GOTO 100                              INT07580
      IF(NX.LT.3 .OR. ITR.NE.3) GO TO 100                               INT07590
      CALL TRNS(ICODE)                                                  INT07600
      IF(ICODE.EQ.1) GOTO 20                                            INT07610
100   IF(NX .NE. NSS) GOTO 120                                          INT07620
C                                                                       INT07630
C   STORE PROFILES AT THE STATION NS FOR INVERSE B. L.                  INT07640
C   CALCULATION                                                         INT07650
C                                                                       INT07660
      DO 110 J = 1 , NPT                                                INT07670
      FS(J) = F(J,2)                                                    INT07680
      US(J) = U(J,2)                                                    INT07690
      VS(J) = V(J,2)                                                    INT07700
      WS(J) = W(J,2)                                                    INT07710
      BS(J) = B(J,2)                                                    INT07720
110   CONTINUE                                                          INT07730
120   IF ( NX .LT. NSS ) GOTO 10                                        INT07740
C     IF (ICYCLE .NE. 1) GO TO 130                                      INT07750
      IF ( NX .GE. NS )  GOTO 130                                       INT07760
```

```
      IF (NX .LT. NXT) GO TO 10                                          INT0777
      CALL HEADER( TITLE,SURFID,ISTRP )                                  INT0778
      WRITE(6, 170 ) (M,X(M),CF(M),DLS(M),UE(M),P2  (M),THT(M),          INT0779
     +             D(M), ALFAS(M), ITP(M),NPSTR(M),M=1,NXT)              INT0780
130   DO 140 I = 1 , NXT                                                 INT0781
140   DB(I) = D(I)                                                       INT0782
      NS = NSS                                                           INT0783
      NX = NS                                                            INT0784
      NP = NPSTR(NX)                                                     INT0785
      DO 150 J = 1 , NPT                                                 INT0786
      F(J,2) = FS(J)                                                     INT0787
      U(J,2) = US(J)                                                     INT0788
      V(J,2) = VS(J)                                                     INT0789
      W(J,2) = WS(J)                                                     INT0790
      B(J,2) = BS(J)                                                     INT0791
150   CONTINUE                                                           INT0792
155   INVRS = NS + 1                                                     INT0793
C                                                                        INT0794
C   CALCULATION SHIFTS TO USING PHYSICAL COORDINATES                     INT0795
      CALL MAIN2(ITR,ISWPT,SURFID)                                       INT0796
C                                                                        INT0797
C     PASS DELTA-STAR BACK TO MAIN PROG.                                 INT0798
C                                                                        INT0799
      DO 158 I = 1,NXT                                                   INT0800
      DELS(I) = DLS(I)                                                   INT0801
158   CONTINUE                                                           INT0802
      RETURN                                                             INT0803
C                                                                        INT0804
C   ------------------------------------------------------------         INT0805
C                                                                        INT0806
160   FORMAT(1H0,' ** ITERATIONS EXCEEDED ITMAX AT NX = ',I5/            INT0807
     +            1H ,' ** CALCULATIONS STOP. ** ')                      INT0808
170   FORMAT(1H0,' ** SUMMARY OF STANDARD B. L. SOLUTIONS. **'/          INT0809
     +            1H0,4X,2HNX,7X,1HX,12X,2HCF,11X,3HDLS,12X,2HUE,       INT0810
     +            12X,2HP2,11X,3HTHT,13X,1HD,10X,4HALFA,6X,2HIT,2X,2HNP/INT0811
     +            (1H ,3X,I3,F10.5,2X,7E14.5,2I4))                       INT0812
180   FORMAT(1H0,34H FLOW SEPARATES. D IS EXTRAPOLATED/                  INT0813
     +            1H0,3X,3H NX,7X,1HX,13X,1HD/)                          INT0814
190   FORMAT(1H ,3X,I3,F10.5,2X,E14.5)                                   INT0815
      END                                                                INT0816
C          DATA SET KCBCCALCIJ AT LEVEL 001 AS OF 08/24/84               INT0817
C          DATA SET KCBCCALCIJ AT LEVEL 001 AS OF 08/24/84               INT0818
C          DATA SET KCBCCALCIJ AT LEVEL 005 AS OF 04/05/84               INT0819
      SUBROUTINE CALCIJ ( IL, L0)                                        INT0820
C                                                                        INT0821
C   CALCULATE HILBERT INTEGRAL COEFFS                                    INT0822
C                                                                        INT0823
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI       INT0824
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                     INT0825
     +                    ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT       INT0826
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH              INT0827
      DIMENSION    E(2)                                                  INT0828
C                                                                        INT0829
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  INT0830
C                                                                        INT0831
      PI      = 3.14159265                                               INT0832
```

```
      PI       = PI*SQRT(RL)                                              INT08330
      IL1      = IL  - 1                                                  INT08340
      DO 65 I = 2, IL1                                                    INT08350
      E (1)  = 0.                                                         INT08360
      L       = L0 + I                                                    INT08370
      DO 60 J = 2, IL                                                     INT08380
      J1       = J - 1                                                    INT08390
      K        = J + L0                                                   INT08400
      DX1      = X(L) - X(K)                                              INT08410
      DX2      = X(K) - X(K-1)                                            INT08420
      DX3      = X(L) - X(K-1)                                            INT08430
      IF ( J .EQ. I ) GO TO 30                                            INT08440
      IF ( J .EQ. (I+1) ) GO TO 40                                        INT08450
C                                                                         INT08460
C     J .NE. I OR I+1                                                     INT08470
C                                                                         INT08480
      E (2)    = ( 1.0/DX2 ) * ALOG( ABS( DX3 / DX1 ) )                   INT08490
      GO TO 50                                                            INT08500
C                                                                         INT08510
C     J .EQ. I                                                            INT08520
C                                                                         INT08530
30    R1       = ( X(K+1)-X(K) ) / (X(K+1)-X(K-1) )                       INT08540
      E (2)    = ( R1 * ALOG( ABS( DX3 / ( X(L)-X(K+1)) ) ) + 2.0 ) / DX2INT08550
      GO TO 50                                                            INT08560
C                                                                         INT08570
C     J .EQ. I+1                                                          INT08580
C                                                                         INT08590
40    R1       = ( X(K-1)-X(K-2) ) / ( X(K)-X(K-2) )                      INT08600
      E (2)    = ( R1 * ALOG( ABS( (X(L)-X(K-2)) / DX1 ) ) - 2.0 ) / DX2 INT08610
C                                                                         INT08620
50    CONTINUE                                                            INT08630
      C (J1,I)= ( E(1) - E(2) ) / PI                                      INT08640
      E(1)     = E (2)                                                    INT08650
60    CONTINUE                                                            INT08660
      E (2)    = 0.                                                       INT08670
      J1       = IL                                                       INT08680
      K        = K + 1                                                    INT08690
      C (J1,I)= E(1) / PI                                                 INT08700
65    CONTINUE                                                            INT08710
C                                                                         INT08720
C                                                                         INT08730
      RETURN                                                              INT08740
      END                                                                 INT08750
C          DATA SET KCBCCOEF   AT LEVEL 001 AS OF 08/24/84                INT08760
C          DATA SET KCBCCOEF   AT LEVEL 001 AS OF 08/24/84                INT08770
C          DATA SET KCBCCOEF   AT LEVEL 007 AS OF 04/05/84                INT08780
      SUBROUTINE COEF(GAMMA1,GAMMA2)                                      INT08790
C                                                                         INT08800
C   CALCULATE COEFFS OF B. L. FINITE-DIFFERENCE EQS. IN                   INT08810
C   SEMI-TRANSF VARIABLES( AFTER SWITCHING).                              INT08820
C                                                                         INT08830
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                      INT08840
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)          INT08850
      COMMON /BLC6/ S1(101),S2(101),S3(101),S4(101),S5(101),S6(101),      INT08860
     +                   S7(101),S8(101),R1(101),R2(101),R3(101),R4(101)INT08870
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI        INT08880
```

```
      COMMON /GRD / ETA(101),DETA(101),A(101)                          INT0889
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH            INT0890
C                                                                      INT0891
C   ---------------------------------------------------------------    INT0892
C                                                                      INT0893
      P1H = 0.5 * P1(NX)                                               INT0894
      DO 100 J= 2,NP                                                   INT0895
      FLARE  = 1.0                                                     INT0896
      FB     = 0.5*(F(J,2) + F(J-1,2))                                 INT0897
      UB     = 0.5*(U(J,2) + U(J-1,2))                                 INT0898
      FVB    = 0.5*(F(J,2)*V(J,2)+F(J-1,2)*V(J-1,2))                   INT0899
      IF(UB .LT. 0.0) FLARE = 0.0                                      INT0900
      VB     = 0.5*(V(J,2) + V(J-1,2))                                 INT0901
      USB    = 0.5*(U(J,2)**2 + U(J-1,2)**2)                           INT0902
      WSB    = 0.5*(W(J,2)**2 + W(J-1,2)**2)                           INT0903
      DERBV  =(B(J,2)*V(J,2) - B(J-1,2)*V(J-1,2))/DETA(J-1)            INT0904
      FB4    = 0.5*(F(J,1) + F(J-1,1))                                 INT0905
      VB4    = 0.5*(V(J,1) + V(J-1,1))                                 INT0906
      USB4   = 0.5*(U(J,1)**2 + U(J-1,1)**2)                           INT0907
      WSB4   = 0.5*(W(J,1)**2 + W(J-1,1)**2)                           INT0908
      FVB4   = 0.5*(F(J,1)*V(J,1)+F(J-1,1)*V(J-1,1))                   INT0909
      DERBV4 = (B(J,1)*V(J,1) - B(J-1,1)*V(J-1,1))/DETA(J-1)           INT0910
      S1(J)  = CELH*(FB - FB4) + P1H*F(J,2) + B(J,2)/DETA(J-1)         INT0911
      S2(J)  = CELH*(FB - FB4) + P1H*F(J-1,2) - B(J-1,2)/DETA(J-1)     INT0912
      S3(J)  = CELH*(VB + VB4) + P1H*V(J,2)                            INT0913
      S4(J)  = CELH*(VB + VB4) + P1H*V(J-1,2)                          INT0914
      S5(J)  = -CEL*FLARE*U(J,2)                                       INT0915
      S6(J)  = -CEL*FLARE*U(J-1,2)                                     INT0916
      S7(J)  = CEL*W(J,2)                                              INT0917
      S8(J)  = CEL*W(J-1,2)                                            INT0918
C                                                                      INT0919
      CRB    = -DERBV4 + CEL*WSB4 - CEL*FLARE*USB4 - P1(NX)*FVB4       INT0920
      R2(J)  = CRB - (DERBV - CEL*FLARE*USB  + CEL*(VB+VB4)*(FB-FB4) + INT0921
     +                    CEL*WSB + P1(NX)*FVB)                        INT0922
      R1(J)  = F(J-1,2) - F(J,2) + DETA(J-1)*UB                        INT0923
      R3(J-1)= U(J-1,2) - U(J,2) + DETA(J-1)*VB                        INT0924
      R4(J-1)= W(J-1,2) - W(J,2)                                       INT0925
100   CONTINUE                                                         INT0926
C                                                                      INT0927
C   BOUNDARY CONDITIONS                                                INT0928
C                                                                      INT0929
      R1(1)  = 0.0                                                     INT0930
      R2(1)  = 0.0                                                     INT0931
      R4(NP) = 0.0                                                     INT0932
      IF(NX .GE. INVRS) GO TO 120                                      INT0933
      GAMMA1 = 0.0                                                     INT0934
      GAMMA2 = 1.0                                                     INT0935
      R3(NP) = 0.0                                                     INT0936
      RETURN                                                           INT0937
120   CONTINUE                                                         INT0938
      CII    = C(NX,NX) * SQRT(X(NX))                                  INT0939
      GAMMA1 = 1.0                                                     INT0940
      GAMMA2 = (1.0 - CII*ETA(NP))/CII                                 INT0941
      R3(NP) = (GI + CII*(ETA(NP)*W(NP,2) - F(NP,2)) -W(NP,2))/CII     INT0942
C                                                                      INT0943
      RETURN                                                           INT0944
```

```
      END                                                               INT09450
C          DATA SET KCBCCOEFTR AT LEVEL 001 AS OF 08/24/84              INT09460
C          DATA SET KCBCCOEFTR AT LEVEL 001 AS OF 08/24/84              INT09470
C          DATA SET KCBCCOEFTR AT LEVEL 004 AS OF 02/21/84              INT09480
      SUBROUTINE COEFTR                                                 INT09490
C                                                                       INT09500
C   CALCULATE COEFFS. OF B. L. FINITE-DIFFERENCE EQS.                   INT09510
C   IN TRANSFORMED VARIABLES( BEFORE SWITCHING).                        INT09520
C                                                                       INT09530
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT09540
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT09550
      COMMON /BLC6/ S1(101),S2(101),S3(101),S4(101),S5(101),S6(101),    INT09560
     +                    S7(101),S8(101),R1(101),R2(101),R3(101),R4(101)INT09570
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT09580
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT09590
C                                                                       INT09600
C   ---------------------------------------------------                 INT09610
C                                                                       INT09620
C                                                                       INT09630
      DO 100 J= 2,NP                                                    INT09640
      FB     = 0.5*(F(J,2) + F(J-1,2))                                  INT09650
      UB     = 0.5*(U(J,2) + U(J-1,2))                                  INT09660
      VB     = 0.5*(V(J,2) + V(J-1,2))                                  INT09670
      USB    = 0.5*(U(J,2)**2 + U(J-1,2)**2)                            INT09680
      DERBV = (B(J,2)*V(J,2) - B(J-1,2)*V(J-1,2))/DETA(J-1)             INT09690
      FVB = 0.5*(F(J,2)*V(J,2) + F(J-1,2)*V(J-1,2))                     INT09700
      FVB4 = 0.5*(F(J,1)*V(J,1) + F(J-1,1)*V(J-1,1))                    INT09710
      FB4 = 0.5*(F(J,1) + F(J-1,1))                                     INT09720
      VB4 = 0.5*(V(J,1) + V(J-1,1))                                     INT09730
      USB4   = 0.5*(U(J,1)**2 + U(J-1,1)**2)                            INT09740
      DERBV4 = (B(J,1)*V(J,1) - B(J-1,1)*V(J-1,1))/DETA(J-1)            INT09750
C                                                                       INT09760
      S1(J) = CELH*(FB-FB4) + 0.5*P1(NX)*F(J,2) + B(J,2)/DETA(J-1)      INT09770
      S2(J) = CELH*(FB-FB4) + 0.5*P1(NX)*F(J-1,2) -B(J-1,2)/            INT09780
     +             DETA(J-1)                                            INT09790
      S3(J) = CELH*(VB + VB4) + 0.5*P1(NX)*V(J,2)                       INT09800
      S4(J) = CELH*(VB + VB4) + 0.5*P1(NX)*V(J-1,2)                     INT09810
      S5(J) = -(CEL+P2(NX))*U(J,2)                                      INT09820
      S6(J) = -(CEL+P2(NX))*U(J-1,2)                                    INT09830
C                                                                       INT09840
      CLB = DERBV4  + P1(NX-1)*FVB4 - P2(NX-1)*USB4 + P2(NX-1)          INT09850
      CRB = -CLB - CEL*USB4 - P2(NX)                                    INT09860
      R2(J) = CRB - (DERBV + P1(NX)*FVB- (CEL+P2(NX))*USB  + CEL*       INT09870
     +             (VB + VB4)*(FB - FB4))                               INT09880
C                                                                       INT09890
      R1(J)= F(J-1,2) - F(J,2) + DETA(J-1)*UB                           INT09900
      R3(J-1)= U(J-1,2) - U(J,2) + DETA(J-1)*VB                         INT09910
100   CONTINUE                                                          INT09920
      R1(1)  = 0.0                                                      INT09930
      R2(1)  = 0.0                                                      INT09940
      R3(NP) = 0.0                                                      INT09950
      RETURN                                                            INT09960
      END                                                               INT09970
C          DATA SET KCBCCOMPBL AT LEVEL 001 AS OF 08/24/84              INT09980
C          DATA SET KCBCCOMPBL AT LEVEL 001 AS OF 08/24/84              INT09990
```

```
C           DATA SET KCBCCOMPBL AT LEVEL 010 AS OF 08/24/84                 INT1000
C                                                                           INT1001
      SUBROUTINE COMPBL(CASEID,XP,YP,VT,S,DLSP,DLS,DBPP,NBL,ITRI,XCTRI,     INT1002
     +                          RN,NT,ISWPT)                              INT1003
C                                                                           INT1004
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                        INT1005
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI          INT1006
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                        INT1007
     +                   ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT          INT1008
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH                 INT1009
      COMMON /BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),           INT1010
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)           INT1011
      COMMON /BLOW/ VN(100)                                                 INT1012
      COMMON /ISURF/ ISF                                                    INT1013
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                     INT1014
C                                                                           INT1015
      DIMENSION     XP(100),DLSP(100),YP(100),VT(100),S(100),               INT1016
     +              DBPP(100),DLS(100),CASEID(20)                           INT1017
      DIMENSION     XIN(100,2),YIN(100,2),SI(100,2),VIN(100,2),             INT1018
     +                   DIN(100,2),DELSTR(100,2),DD(100,2),DDD(100,2)      INT1019
      DIMENSION     XB(101),D1(101),D2(101),D3(101),IEND(2)                 INT1020
      DIMENSION     NBL(2),ITRI(2),XCTRI(2)                                 INT1021
      LOGICAL TRFIND                                                        INT1022
      CHARACTER * 4 SURF(4),STITLE(2),SURFID(4)                             INT1023
C                                                                           INT1024
      DATA     SURF   / '    ','R SU','RFAC','E   ' /                       INT1025
      DATA     STITLE / 'UPPE','LOWE' /                                     INT1026
C                                                                           INT1027
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT1028
C                                                                           INT1029
90    FORMAT ( 1H1,5X,'COMPUTING BOUNDARY LAYER USING HILBERT',             INT1030
     +                 ' INTEGRAL',/)                                       INT1031
110   FORMAT ( 1H0,6X,'I',9X,'XP',13X,'YP',13X,'S',14X,'VT',13X,            INT1032
     +             'DBP' / )                                                INT1033
112   FORMAT ( 1H0,6X,'I',4X,'II',3X,'IK',7X,'XIN',12X,'YIN',               INT1034
     +             13X,'SI',12X,'VIN',12X,'DIN' / )                         INT1035
120   FORMAT ( 1H ,5X,I3,5E15.6 )                                           INT1036
122   FORMAT ( 1H ,3X,3(2X,I3),5E15.6 )                                     INT1037
130   FORMAT ( 1H0,5X,'STAGNATION POINT IS FOUND BETWEEN POINT NO. ',       INT1038
     +             I3,' AND POINT NO. ',I3 / )                              INT1039
140   FORMAT ( 1H0,5X,'INTERPOLATED STAGNATION POINT VALUES' /              INT1040
     +             1H0,5X,'S = ',E13.6,2X,'XP = ',E13.6,2X,'YP = ',         INT1041
     +             E13.6,2X,'DBP = ',E13.6,2X,'VT = 0.0' / )                INT1042
150   FORMAT ( 1H0,5X,'TOTAL NUMBER OF UPPER SURFACE POINTS = ',I5,         INT1043
     +             2X,'AND AT LOWER SURFACE = ',I5 / )                      INT1044
160   FORMAT ( 1H0,5X,'UPPER SURFACE DATA' )                                INT1045
170   FORMAT ( 1H0,5X,'LOWER SURFACE DATA' )                                INT1046
180   FORMAT ( 1H0,5X,'UPPER SURFACE CALCULATIONS' / )                      INT1047
182   FORMAT ( 1H0,5X,'LOWER SURFACE CALCULATIONS' / )                      INT1048
190   FORMAT ( 1H0,5X,'RESULTS OF POINT REDISTRIBUTION' / )                 INT1049
200   FORMAT ( 1H0,5X,'TABLE OF DELTA-STARS' / ( 1H ,5X,8E15.6 ) )          INT1050
220   FORMAT (1H0,5X,'TABLE OF BLOWING-VEL.' / (1H ,5X,8E15.6) )            INT1051
210   FORMAT ( 1H0,5X,'NO CHANGE OF SIGN ON VT. CANNOT FIND STAG. PT.' )    INT1052
C                                                                           INT1053
C             -------------- * -------------                                INT1054
C                                                                           INT1055
```

```
C                                                                       INT10560
C     READ ONE STRIP INPUT DATA FROM UNIT NO. 1.  THE ORDER IS          INT10570
C     FROM THE LOWER SURFACE T.E. TO THE UPPER SURFACE T.E.             INT10580
C                                                                       INT10590
      DO 230 I = 1,20                                                   INT10600
      TITLE(I) = CASEID(I)                                              INT10610
230   CONTINUE                                                          INT10620
      RL     = RN                                                       INT10630
      DO 300 I = 1,NT                                                   INT10640
      DBP(I) = DBPP(I)                                                  INT10650
300   CONTINUE                                                          INT10660
C                                                                       INT10670
C     PRINT THE INPUT DATA.                                             INT10680
C                                                                       INT10690
      P2(1)  = 1.0                                                      INT10700
C     WRITE ( 6,90 )                                                    INT10710
C     WRITE ( 6,110 )                                                   INT10720
C     DO 500 I = 1,NT                                                   INT10730
C     WRITE ( 6,120 ) I,XP(I),YP(I),S(I),VT(I),DBP(I)                   INT10740
C500  CONTINUE                                                          INT10750
C                                                                       INT10760
C     FIND STAGNATION POINT                                             INT10770
C                                                                       INT10780
      DO 600 I = 2,NT                                                   INT10790
      VPROD  = VT(I) * VT(I-1)                                          INT10800
      IF ( VPROD .GT. 0. ) GO TO 600                                    INT10810
      IS     = I                                                        INT10820
      IS1    = IS - 1                                                   INT10830
      GO TO 700                                                         INT10840
600   CONTINUE                                                          INT10850
      WRITE ( 6,210 )                                                   INT10860
      STOP 1                                                            INT10870
C                                                                       INT10880
C                                                                       INT10890
C     INTERPOLATE S AT VT = 0.                                          INT10900
C                                                                       INT10910
C     WRITE ( 6,130 ) IS1,IS                                            INT10920
700   DS     = S(IS) - S(IS1)                                           INT10930
      DV     = VT(IS) - VT(IS1)                                         INT10940
      SS     = S(IS) - VT(IS) * ( DS/DV )                               INT10950
      DBB    = DBP(IS) - DBP(IS1)                                       INT10960
      DX     = XP(IS) - XP(IS1)                                         INT10970
      DY     = YP(IS) - YP(IS1)                                         INT10980
      DS1    = SS - S(IS)                                               INT10990
      DBS    = DBP(IS) + DS1 * ( DBB/DS )                               INT11000
      XPS    = XP(IS)  + DS1 * ( DX /DS )                               INT11010
      YPS    = YP(IS)  + DS1 * ( DY /DS )                               INT11020
C     WRITE ( 6,140 ) SS,XPS,YPS,DBS                                    INT11030
C                                                                       INT11040
C     IU IS THE TOTAL UPPER SURFACE POINTS. + STAG. PT.                 INT11050
C     IL IS THE TOTAL LOWER SURFACE POINTS  + STAG. PT.                 INT11060
C                                                                       INT11070
      IU     = NT - IS + 2                                              INT11080
      IL     = IS                                                       INT11090
C     WRITE ( 6,150 ) IU,IL                                             INT11100
C                                                                       INT11110
```

```
C     GROUP THE DATA FOR EACH SURFACE. FIRST, UPPER.                    INT11120
C                                                                       INT11130
      DO 1200 L = 1,2                                                   INT11140
      GO TO ( 800,900 ), L                                              INT11150
C                                                                       INT11160
C     L = 1 IS UPPER SURFACE                                            INT11170
C     L = 2 IS LOWER SURFACE                                            INT11180
C                                                                       INT11190
800   M1     = IS                                                       INT11200
      M2     = NT                                                       INT11210
      IEND(L) = IU                                                      INT11220
      GO TO 1000                                                        INT11230
900   M1     = 1                                                        INT11240
      M2     = IL-1                                                     INT11250
      IEND(L) = IL                                                      INT11260
C                                                                       INT11270
1000  I = 1                                                             INT11280
      XIN(I,L) = XPS                                                    INT11290
      YIN(I,L) = YPS                                                    INT11300
      SI (I,L) = 0.                                                     INT11310
      DIN(I,L) = DBS                                                    INT11320
      VIN(I,L) = 0.                                                     INT11330
      IF ( IP .GE. 1 ) THEN                                             INT11340
      IF ( L .EQ. 1 ) WRITE ( 6,160 )                                   INT11350
      IF ( L .EQ. 2 ) WRITE ( 6,170 )                                   INT11360
      WRITE ( 6,112 )                                                   INT11370
      WRITE ( 6,122 ) I,I,I,XIN(1,L),YIN(1,L),SI(1,L),VIN(1,L),         INT11380
     +                DIN(1,L)                                         INT11390
      END IF                                                            INT11400
      DO 1100 II = M1,M2                                                INT11410
      I      = I + 1                                                    INT11420
      IK     = II                                                       INT11430
      IF ( L .EQ. 2 ) IK = IL - II                                      INT11440
      XIN(I,L) = XP(IK)                                                 INT11450
      YIN(I,L) = YP(IK)                                                 INT11460
      SI (I,L) = S(IK)-SS                                               INT11470
      IF ( L .EQ. 2 ) SI(I,L) = SS-S(IK)                                INT11480
      VIN(I,L) = ABS(VT(IK))                                            INT11490
      DIN(I,L) = DBP(IK)                                                INT11500
      IF(IP .GE. 1)WRITE ( 6,122 ) I,II,IK,XIN(I,L),YIN(I,L),SI(I,L),   INT11510
     +               VIN(I,L),DIN(I,L)                                  INT11520
1100  CONTINUE                                                          INT11530
C                                                                       INT11540
1200  CONTINUE                                                          INT11550
C                                                                       INT11560
C     RE-DISTRIBUTE POINTS ON EACH SURFACE TO A DENSER NUMBER.          INT11570
C                                                                       INT11580
C     WRITE ( 6,90 )                                                    INT11590
      DO 2000 ISF = 1,2                                                 INT11600
      NN     = IEND(ISF)                                                INT11610
      ITR    = ITRI(ISF)                                                INT11620
      NXT    = NBL(ISF)                                                 INT11630
      XCTR   = XCTRI(ISF)                                               INT11640
      SURF  (1) = STITLE(ISF)                                           INT11650
      ICC    = 1                                                        INT11660
      DO 1220 J = 1,4                                                   INT11670
```

```
      SURFID(J) = SURF(J)                                                INT11680
1220  CONTINUE                                                           INT11690
C     IF ( ISF .EQ. 1 ) WRITE ( 6,180 )                                  INT11700
C     IF ( ISF .EQ. 2 ) WRITE ( 6,182 )                                  INT11710
      SCALE  = SI(NN,ISF)                                                INT11720
C                                                                        INT11730
      CALL BLGRID ( NXT,XB,D1,D2 )                                       INT11740
C                                                                        INT11750
      DO 1300 I = 1,NXT                                                  INT11760
1300  X (I)  = XB(I) * SCALE                                             INT11770
C                                                                        INT11780
C     INTERPOLATE S,VT,X,Y,D INTO THE NEW DISTRIBUTION.                  INT11790
C                                                                        INT11800
      CALL SMFIT ( 1,NN,SI(1,ISF),VIN(1,ISF),D1,2 )                      INT11810
      CALL SMFIT ( 1,NN,SI(1,ISF),DIN(1,ISF),D1,2 )                      INT11820
      CALL DIFF3 ( NN,SI(1,ISF),VIN(1,ISF),D1,D2,D3,0 )                  INT11830
      CALL INTRP3( NN,SI(1,ISF),VIN(1,ISF),D1,D2,D3,NXT,X,UE )           INT11840
      CALL DIFF3 ( NN,SI(1,ISF),DIN(1,ISF),D1,D2,D3,0 )                  INT11850
      CALL INTRP3( NN,SI(1,ISF),DIN(1,ISF),D1,D2,D3,NXT,X,DB )           INT11860
      CALL DIFF3 ( NN,SI(1,ISF),XIN(1,ISF),D1,D2,D3,0 )                  INT11870
      CALL INTRP3( NN,SI(1,ISF),XIN(1,ISF),D1,D2,D3,NXT,X,XC )           INT11880
      CALL DIFF3 ( NN,SI(1,ISF),YIN(1,ISF),D1,D2,D3,0 )                  INT11890
      CALL INTRP3( NN,SI(1,ISF),YIN(1,ISF),D1,D2,D3,NXT,X,YC )           INT11900
      IF ( IP .GE. 1 ) THEN                                              INT11910
      WRITE ( 6,190 )                                                    INT11920
      WRITE ( 6,110 )                                                    INT11930
      DO 1320 I = 1,NXT                                                  INT11940
      WRITE ( 6,120 ) I,XC(I),YC(I),X(I),UE(I),DB(I)                     INT11950
1320  CONTINUE                                                           INT11960
      END IF                                                             INT11970
C                                                                        INT11980
C     INPUT TO THE B.L. PROGRAM X,UE,DB,XC,YC ARE NOW DEFINED.           INT11990
C                                                                        INT12000
      DO 1350 I = 1,NXT                                                  INT12010
      DBP(I) = DB(I)                                                     INT12020
      D(I)   = DB(I)                                                     INT12030
1350  CONTINUE                                                           INT12040
C                                                                        INT12050
      CALL BL2D( ITR,ISWPT,SURFID)                                       INT12060
C                                                                        INT12070
      CALL DIFF3 ( NXT,X,DELS,D1,D2,D3,0 )                               INT12080
      CALL INTRP3( NXT,X,DELS,D1,D2,D3,NN,SI(1,ISF),DELSTR(1,ISF) )      INT12090
      CALL DIFF3(NXT,X,D,D1,D2,D3,0)                                     INT12100
      CALL INTRP3(NXT,X,D,D1,D2,D3,NN,SI(1,ISF),DD(1,ISF))               INT12110
      CALL DIFF3(NN,SI(1,ISF),DD(1,ISF),DDD(1,ISF),D2,D3,0)              INT12120
      TRFIND(ISF) = .FALSE.                                              INT12130
      IF(ITR .EQ. 3 .AND. NTR .LE. NXT) THEN                             INT12140
      XCTRS(ISF) = XCTR                                                  INT12150
      TRFIND(ISF) = .TRUE.                                               INT12160
      END IF                                                             INT12170
C                                                                        INT12180
2000  CONTINUE                                                           INT12190
C                                                                        INT12200
C     PUT THE TWO SURFACES DELTA-STARS BACK TO ONE STRIP                 INT12210
C                                                                        INT12220
      DELSTR(1,2) = 0.5*(DELSTR(2,1)+DELSTR(2,2))                        INT12230
```

```
      DELSTR(1,1) = DELSTR(1,2)                                          INT1224(
      DD(1,2)  = 0.5 * (DD(2,1) + DD(2,2))                               INT1225(
      DD(1,1)  = DD(1,2)                                                 INT1226(
      DDD(1,2) = 0.5 * (DDD(2,1) + DDD(2,2)) /SQRT(RL)                   INT1227(
      DDD(1,1) = DDD(1,2)                                                INT1228(
      IL = IEND(2)                                                       INT1229(
      I      = IL                                                        INT1230(
      DO 2100 II = 2,IL                                                  INT1231(
      I      = I-1                                                       INT1232(
      DLS(I) = DELSTR(II,2)                                              INT1233(
      VN(I) = DDD(II,2)/SQRT(RL)                                         INT1234(
      DBPP(I) = DD(II,2)                                                 INT12350
2100  CONTINUE                                                           INT12360
C                                                                        INT12370
      I      = IL-1                                                      INT12380
      IU     = IEND(1)                                                   INT12390
      DO 2200 II=2,IU                                                    INT12400
      I      = I + 1                                                     INT12410
      DLS(I) = DELSTR(II,1)                                              INT12420
      VN(I) = DDD(II,1)/SQRT(RL)                                         INT12430
      DBPP(I) = DD(II,1)                                                 INT12440
2200  CONTINUE                                                           INT12450
C     WRITE ( 6,200 ) ( DLS(I), I=1,NT )                                 INT12460
C     WRITE ( 6,220) (VN(I) , I=1,NT)                                    INT12470
C                                                                        INT12480
      RETURN                                                             INT12490
      END                                                                INT12500
C          DATA SET KCBCCOMPGI AT LEVEL 001 AS OF 08/24/84               INT12510
C          DATA SET KCBCCOMPGI AT LEVEL 001 AS OF 08/24/84               INT12520
C          DATA SET KCBCCOMPGI AT LEVEL 003 AS OF 08/24/84               INT12530
      SUBROUTINE COMPGI                                                  INT12540
C                                                                        INT12550
C   CALCULATE GI                                                         INT12560
C                                                                        INT12570
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                     INT12580
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI       INT12590
C                                                                        INT12600
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT12610
C                                                                        INT12620
      SUM2     = 0.0                                                     INT12630
      N1       = NX + 1                                                  INT12640
      N2       = NXT                                                     INT12650
      DO 180 K = N1,N2                                                   INT12660
      SUM2     = SUM2 + C(K,NX)* (D(K) - DBP(K))                         INT12670
180   CONTINUE                                                           INT12680
C                                                                        INT12690
      N1       = 2                                                       INT12700
      N2       = NX-1                                                    INT12710
      SUM1     = 0.                                                      INT12720
      DO 260 K = N1,N2                                                   INT12730
      SUM1     = SUM1 + C(K,NX)* (D(K) - DBP(K))                         INT12740
260   CONTINUE                                                           INT12750
      GI       = UE0(NX) + SUM1 + SUM2 - C(NX,NX) * DBP(NX)              INT12760
C                                                                        INT12770
      RETURN                                                             INT12780
```

```
      END                                                               INT12790
C          DATA SET KCBCDIFF3  AT LEVEL 001 AS OF 08/24/84              INT12800
C          DATA SET KCBCDIFF3  AT LEVEL 001 AS OF 08/24/84              INT12810
C          DATA SET KCBCDIFF3  AT LEVEL 002 AS OF 04/05/84              INT12820
      SUBROUTINE DIFF3 (N,X,F,FP,FPP,FPPP,IEND)                         INT12830
C                                                                       INT12840
C     DETERMINES THE DERIVATIVE OF THE INPUT FUNCTION AT THE INPUT PTS. INT12850
C                                                                       INT12860
      DIMENSION X(101),F(101),FP(101),FPP(101),FPPP(101)                INT12870
C                                                                       INT12880
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   INT12890
C                                                                       INT12900
C                                                                       INT12910
C     FIRST DERIVATIVES USING WEIGHTED ANGLES                           INT12920
C                                                                       INT12930
      N1=N-1                                                            INT12940
      DX=X(2)-X(1)                                                      INT12950
      DF=F(2)-F(1)                                                      INT12960
      ANG2= ATAN2(DF,DX)                                                INT12970
      DL2=DX                                                            INT12980
C                                                                       INT12990
      DO 10 I=2,N1                                                      INT13000
      ANG1=ANG2                                                         INT13010
      DL1=DL2                                                           INT13020
      I1=I+1                                                            INT13030
      DX=X(I1)-X(I)                                                     INT13040
      DF=F(I1)-F(I)                                                     INT13050
      ANG2= ATAN2(DF,DX)                                                INT13060
      DL2=DX                                                            INT13070
      ANG=(DL2*ANG1+DL1*ANG2)/(DL1+DL2)                                 INT13080
      FP(I)= TAN(ANG)                                                   INT13090
C                                                                       INT13100
      IF (I.NE.2) GO TO 10                                              INT13110
      ANGI = ANG                                                        INT13120
      ANG1I = ANG1                                                      INT13130
      DLI = DL1                                                         INT13140
C                                                                       INT13150
10    CONTINUE                                                          INT13160
      ANGF = ANG                                                        INT13170
      ANG2F = ANG2                                                      INT13180
      DLF = DL2                                                         INT13190
      IEND1 = IEND + 1                                                  INT13200
      GO TO (11,12,13), IEND1                                           INT13210
C                                                                       INT13220
C     IEND = 0 , EXTRAPOLATE FOR END VALUES                             INT13230
C                                                                       INT13240
11    FP(1) = 2.*(F(2)-F(1 ))/DLI - FP(2)                               INT13250
      FP(N) = 2.*(F(N)-F(N1))/DLF - FP(N1)                              INT13260
      GO TO 20                                                          INT13270
C                                                                       INT13280
C     IEND = 1, DERIVATIVES ARE CONTINUOUS ACROSS ENDS                  INT13290
C                                                                       INT13300
12    ANG = (DLI*ANG2F + DLF*ANG1I) / (DLI + DLF)                       INT13310
      FP(1) =  TAN(ANG)                                                 INT13320
      FP(N) = FP(1)                                                     INT13330
```

```
      GO TO 20                                                          INT1334
C                                                                       INT1335
C  IEND = 2, IF FIRST DERIVATIVE .LT. 0.0 RESET TO ZERO                 INT1336
C                                                                       INT1337
13    CONTINUE                                                          INT1338
      FP(1) = 2.*(F(2)-F(1 ))/DLI - FP(2)                               INT1339
      IF (FP (1) .LT. 0.0) FP (1) = 0.0                                 INT1340
      FP(N) = 2.*(F(N)-F(N1))/DLF - FP(N1)                              INT1341
C                                                                       INT1342
C     SECOND + THIRD DERIVATIVES USING CUBIC FITS                       INT1343
C                                                                       INT1344
20    DO 30 I=2,N1                                                      INT1345
      I1     = I - 1                                                    INT1346
      I2     = I + 1                                                    INT1347
      DX1    = X (I1) - X (I)                                           INT1348
      DX2    = X (I2) - X (I)                                           INT1349
      DF1    = 2.0 * ((F (I1) - F (I)) / DX1 - FP (I)) / DX1            INT1350
      DF2    = 2.0 * ((F (I2) - F (I)) / DX2 - FP (I)) / DX2            INT1351
      FPPP(I)= 3.0 * (DF1 - DF2) / (DX1 - DX2)                          INT1352
      FPP (I)= DF1 - DX1 * FPPP (I) / 3.0                               INT1353
30    CONTINUE                                                          INT1354
      FPPP(1)= FPPP (2)                                                 INT1355
      FPPP(N)= FPPP (N1)                                                INT1356
      FPP (1)= FPP (2 ) + (X (1) - X (2 )) * FPPP (2 )                  INT1357
      FPP (N)= FPP (N1) + (X (N) - X (N1)) * FPPP (N1)                  INT1358
C                                                                       INT1359
      RETURN                                                            INT1360
      END                                                               INT1361
C          DATA SET KCBCEDDY   AT LEVEL 001 AS OF 08/24/84              INT1362
C          DATA SET KCBCEDDY   AT LEVEL 001 AS OF 08/24/84              INT1363
C          DATA SET KCBCEDDY   AT LEVEL 003 AS OF 04/05/84              INT1364
      SUBROUTINE EDDY                                                   INT1365
C                                                                       INT1366
C   CALCULATE EDDY VISCOSITY USING C .S. TWO-LAYERS EDDY                INT1367
C   VISCOSITY FORMULA                                                   INT1368
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT1369
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT1370
      COMMON/BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI       INT1371
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100),                   INT1372
     +                    ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT       INT1373
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT1374
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT1375
      DIMENSION FINT(101)                                               INT1376
C                                                                       INT1377
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT1378
C                                                                       INT1379
      JO=1                                                              INT1380
      UDEL=0.995*U(NP,2)                                                INT1381
      DO 10 J=1,NP                                                      INT1382
      IF(U(J,2).LT.UDEL) JJ=J                                           INT1383
10    IF(U(J,2).LT.0.0)  JO=J                                           INT1384
      EDEL=ETA(JJ)+(ETA(JJ+1)-ETA(JJ))/(U(JJ+1,2)-U(JJ,2))              INT1385
     +     *(UDEL-U(JJ,2))                                              INT1386
      DO 15 J=1,NP                                                      INT1387
      ETADEL=ETA(J)/EDEL                                                INT1388
```

```
      IF(ETADEL.GT.1.0) ETADEL=1.0                                      INT13890
15    FINT(J)=1.0/(1.0+5.5*ETADEL**6)                                   INT13900
      CALL AMEAN(1,JJ,ETA,FINT,2)                                       INT13910
      RU = RL                                                           INT13920
      IF (IT .GT. 1) GO TO 20                                           INT13930
      ALFAS(NX) = ALFAS(NX-1)                                           INT13940
      FFS(NX) = FFS(NX-1)                                               INT13950
      RTS(NX) = RTS(NX-1)                                               INT13960
C                                                                       INT13970
      GMTR   = GMTRS(NX)                                                INT13980
      IF (NX .LE. NS) RU = RL * UE(NX)                                  INT13990
      RL2    = SQRT(RU * X(NX))                                         INT14000
      RL4    = SQRT(RL2)                                                INT14010
      RL216  = 0.16 * RL2                                               INT14020
20    VMAX   = 0.5 * (V(1,2) + V(1,1))                                  INT14030
      DO 30 J=2,NP                                                      INT14040
      VM     = 0.5 * (V(J,2)+V(J,1))                                    INT14050
      IF(VM .GT. VMAX) VMAX = VM                                        INT14060
30    CONTINUE                                                          INT14070
      IF (IEDY .EQ. 0) GO TO 35                                         INT14080
      IF (IT .LE. 1 .OR. GMTR .LT. 0.85 .OR. NX .LE. NTR+3 )            INT14090
     +     GO TO 35                                                     INT14100
C                                                                       INT14110
C   MODIFY ALFA USING SIMPSON'S ARGUMENTS                               INT14120
C     CALL SMPSON                                                       INT14130
35    ALFA = ALFAS(NX)                                                  INT14140
      EDVO   = ALFA * RL2 * GMTR * (U(NP,2)*ETA(NP) - F(NP,2))          INT14150
      DO 40 J=2,NP                                                      INT14160
      JJ     = J                                                        INT14170
      YBA    = RL4*SQRT(VMAX)/26.0*ETA(J)                               INT14180
      EL     = 1.0                                                      INT14190
      IF(YBA .LT. 10.0) EL = 1.0 - EXP(-YBA)                            INT14200
      EDVI   = RL216 * GMTR * (EL*ETA(J))**2 * ABS(V(J,2))              INT14210
      IF(EDVI .GT. EDVO) GO TO 70                                       INT14220
      B(J,2) = 1.0 + EDVI*FINT(J)                                       INT14230
      IF(B(J,2) .LT. B(J-1,2)) B(J,2) = B(J-1,2)                        INT14240
C     B(J,2) = 1.0 + EDVI                                               INT14250
40    CONTINUE                                                          INT14260
      JM     = 2                                                        INT14270
      BM     = B(2,2)                                                   INT14280
      DO 50 J=2,NP                                                      INT14290
      IF(BM.GT.B(J,2)) GOTO 50                                          INT14300
      JM     = J                                                        INT14310
      BM     = B(J,2)                                                   INT14320
50    CONTINUE                                                          INT14330
      GOTO 90                                                           INT14340
70    DO 80 J=JJ,NP                                                     INT14350
80    B(J,2) = 1.0 + EDVO*FINT(J)                                       INT14360
C  80 B(J,2) = 1.0 +EDVO                                                INT14370
C                                                                       INT14380
90    CONTINUE                                                          INT14390
      B(1,2) = 1.0                                                      INT14400
C                                                                       INT14410
      JJ     = 1                                                        INT14420
      DO 100 J=2,NP                                                     INT14430
```

```
      IF(U(J,2) .LT. 0.0) JJ = J                                        INT14440
100   CONTINUE                                                          INT14450
      IF(JJ.EQ.1) GO TO 110                                             INT14460
C                                                                       INT14470
C   IN THE SEPARATED REGION, EDDY VISCOSITY IS SET EQUAL TO             INT14480
C   THAT IN THE PREVOUS STATION TO AVOID NUMERICAL TROUBLE              INT14490
      JJP3   = JJ + 3                                                   INT14500
      JJP3   = MIN(JJP3, NP)                                            INT14510
      CALL AMEAN(1,JJP3,ETA,B(1,2),2)                                   INT14520
110   CALL AMEAN(1,NP,ETA,B(1,2),1)                                     INT14530
C                                                                       INT14540
      RETURN                                                            INT14550
      END                                                               INT14560
C          DATA SET KCBCEXTRAP AT LEVEL 001 AS OF 08/24/84              INT14570
C          DATA SET KCBCEXTRAP AT LEVEL 001 AS OF 08/24/84              INT14580
C          DATA SET KCBCEXTRAP AT LEVEL 008 AS OF 02/13/84              INT14590
      SUBROUTINE EXTRAP(NX,NXTE,X,Y)                                    INT14600
C                                                                       INT14610
C   EXTRAPOLATE DATA USING PARABOLIC FORMULA                            INT14620
      DIMENSION X(101),Y(101)                                           INT14630
C - - - - - - - - - - - - - - - - - - - - - - - - - - - -               INT14640
      Y1     = Y(NX-2)                                                  INT14650
      Y2     = Y(NX-1)                                                  INT14660
      X1     = X(NX-2)                                                  INT14670
      X2     = X(NX-1)                                                  INT14680
      X3     = X(NXTE)                                                  INT14690
      X1     = X1 -X3                                                   INT14700
      X2     = X2 -X3                                                   INT14710
      BB     = (Y1-Y2)/(X1**2 - X2**2)                                  INT14720
      AA     = Y1 - BB * X1**2                                          INT14730
      DO 10 I=NX,NXTE                                                   INT14740
      X1     = X(I) -X3                                                 INT14750
      Y(I)   =  AA + BB * X1**2                                         INT14760
10    CONTINUE                                                          INT14770
      RETURN                                                            INT14780
      END                                                               INT14790
C          DATA SET KCBCFILLUP AT LEVEL 001 AS OF 08/24/84              INT14800
C          DATA SET KCBCFILLUP AT LEVEL 001 AS OF 08/24/84              INT14810
C          DATA SET KCBCFILLUP AT LEVEL 007 AS OF 04/05/84              INT14820
      SUBROUTINE FILLUP(INDEX)                                          INT14830
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT14840
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT14850
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT14860
C                                                                       INT14870
C   ------------------------------------------------------              INT14880
C                                                                       INT14890
      IF(NP.GE.NPT ) RETURN                                             INT14900
      IF(INDEX.EQ.2) GOTO 10                                            INT14910
C                                                                       INT14920
C   DEFINE PROFILES FOR B. L. GROWTH                                    INT14930
      NP1    = NP + 1                                                   INT14940
      NP     = NP + 2                                                   INT14950
      NP     = MIN(NP, NPT)                                             INT14960
      NM     = NP                                                       INT14970
      GOTO 20                                                           INT14980
C                                                                       INT14990
```

```
C    FILL UP PROFILES BEFORE MOVING TO THE NEXT STATION                  INT15000
C                                                                        INT15010
10    NP1    = NP + 1                                                    INT15020
      NM     = NPT                                                       INT15030
20    DO 30 J=NP1,NM                                                     INT15040
      F(J,2) = F(J-1,2) + DETA(J-1)*U(J-1,2)                             INT15050
      U(J,2) = U(J-1,2)                                                  INT15060
      V(J,2) = 0.0                                                       INT15070
      B(J,2) = B(J-1,2)                                                  INT15080
      W(J,2) = W(J-1,2)                                                  INT15090
30    CONTINUE                                                           INT15100
C                                                                        INT15110
      RETURN                                                             INT15120
      END                                                                INT15130
C          DATA SET KCBCHEADER AT LEVEL 001 AS OF 08/24/84               INT15140
C          DATA SET KCBCHEADER AT LEVEL 001 AS OF 08/24/84               INT15150
C          DATA SET KCBCHEADER AT LEVEL 001 AS OF 04/05/84               INT15160
      SUBROUTINE HEADER ( TITLE,SURFID,ISTRP )                           INT15170
      COMMON /ISURF/ ISF                                                 INT15180
C                                                                        INT15190
      DIMENSION     TITLE(20),  SURFID(4)                              INT15200
C                                                                        INT15210
10    FORMAT ( 1H1,20X,20A4 )                                            INT15220
20    FORMAT ( 1H0,15X,'BOUNDARY LAYER CALCULATION FOR ',                INT15230
     +         'UPPER SURFACE ',10X,'ICYCLE=',I5 / 16X,71(1H-) / )       INT15240
30    FORMAT ( 1H0,15X,'BOUNDARY LAYER CALCULATION FOR ',                INT15250
     +         'LOWER SURFACE ',10X,'ICYCLE=',I5 / 16X,71(1H-) / )       INT15260
C                                                                        INT15270
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   INT15280
C                                                                        INT15290
      WRITE ( 6,10 ) TITLE                                               INT15300
      IF(ISF .EQ. 1) WRITE ( 6,20 ) ISTRP                                INT15310
      IF(ISF .EQ. 2) WRITE ( 6,30 ) ISTRP                                INT15320
C                                                                        INT15330
      RETURN                                                             INT15340
      END                                                                INT15350
C          DATA SET KCBCINPUT  AT LEVEL 001 AS OF 08/24/84               INT15360
C          DATA SET KCBCINPUT  AT LEVEL 001 AS OF 08/24/84               INT15370
C          DATA SET KCBCINPUT  AT LEVEL 009 AS OF 08/24/84               INT15380
      SUBROUTINE INPUT(ITR,ISWPT,SURFID)                                 INT15390
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                     INT15400
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)         INT15410
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)              INT15420
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI       INT15430
      COMMON /BONV/ ITMAX,EPSL,EPST,CONV                                 INT15440
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100),                    INT15450
     +                    ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT        INT15460
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH              INT15470
      COMMON /BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),        INT15480
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)        INT15490
      COMMON/TRN/ PGAMTR,OMEGA,RTHETB,RTRANB                             INT15500
      COMMON /ISURF/ ISF                                                 INT15510
      DIMENSION D1(100),D2(100),D3(100)                                  INT15520
      DIMENSION SURFID(4),XCS(100)                                       INT15530
      LOGICAL TRFIND                                                     INT15540
C                                                                        INT15550
```

```
C     ------------------------------------------------------------      INT15560
C                                                                       INT15570
      ITMAX   = 15                                                      INT15580
      EPSL    = 0.0001                                                  INT15590
      EPST    = 0.01                                                    INT15600
      NPT     = 101                                                     INT15610
      ETAE    = 8.0                                                     INT15620
      VGP     = 1.14                                                    INT15630
      DETA1   = 0.01                                                    INT15640
      NSS     = NXT / 4                                                 INT15650
C  SEARCH FOR PRESSURE PEAK AS SWITCH POINT                             INT15660
      UMAX = UE(1)                                                      INT15670
      DO 50 I = 2 , NXT                                                 INT15680
      IF (UE(I) .LE. UMAX) GO TO 55                                     INT15690
      UMAX = UE(I)                                                      INT15700
50    CONTINUE                                                          INT15710
      GO TO 60                                                          INT15720
55    NS = I - 4                                                        INT15730
60    IF (NS .GT. NSS) NS = NSS                                         INT15740
      IF (NS .LT.   3) NS = 3                                           INT15750
C                                                                       INT15760
C   CALCULATE THE PRESSURE PARAMETERS P1 + P2 FOR B. L. CALCULATION     INT15770
C   USING TRANSFORMED COORDINATES                                       INT15780
C                                                                       INT15790
      CALL DIFF3 (NXT, X, UE, D1, D2, D3, 0 )                           INT15800
      DO 65 I = 2,NXT                                                   INT15810
      P2(I)   = X(I) * D1(I) /UE(I)                                     INT15820
      P1(I)   = 0.5 * (1.0 + P2(I))                                     INT15830
65    CONTINUE                                                          INT15840
      P1(1)   = 0.5 * (1.0 + P2(1))                                     INT15850
      XCMIN   = XC(1)                                                   INT15860
      MIN     = 1                                                       INT15870
      DO 70 I=1,NXT                                                     INT15880
      IF(XCMIN.LT.XC(I)) GOTO 70                                        INT15890
      XCMIN   = XC(I)                                                   INT15900
      MIN     = I                                                       INT15910
70    CONTINUE                                                          INT15920
      DO 80 I = 1 , NXT                                                 INT15930
      IF (I .GE. MIN) THEN                                              INT15940
         XCS(I) = XC(I)                                                 INT15950
         ISG(I) = 1                                                     INT15960
      ELSE                                                              INT15970
         XCS(I) = -XCS(I)                                               INT15980
         ISG(I) = -1                                                    INT15990
      END IF                                                            INT16000
80    CONTINUE                                                          INT16010
      INVRS   = NS + 1                                                  INT16020
C                                                                       INT16030
C   SEARCH FOR TRANSITION LOCATION                                      INT16040
      ITRP1   = ITR + 1                                                 INT16050
      GOTO (150, 95, 120, 150), ITRP1                                   INT16060
C                                                                       INT16070
C   TRANSITON LOCATION IS INPUT ( = XCTR)                               INT16080
 95   DO 100 I=1,NXT                                                    INT16090
      IF(XCTR.LT.XCS(I)) GOTO 105                                       INT16100
100   CONTINUE                                                          INT16110
```

```
      NTR     = NXT + 1                                                  INT16120
      GOTO 200                                                           INT16130
105   NTR = I-1                                                          INT16140
      IF (NTR.LT.3) THEN                                                 INT16150
         NTR = 3                                                         INT16160
         XCTR = XC(NTR)                                                  INT16170
      END IF                                                             INT16180
      GOTO 200                                                           INT16190
C                                                                        INT16200
C   TRANSITION LOCATION IS SET AT THE PRESSURE PEAK                      INT16210
C                                                                        INT16220
120   UM      = UE(1)                                                    INT16230
      IM      = 1                                                        INT16240
      DO 75 I = 1,NXT                                                    INT16250
      IF(UM.GT.UE(I)) GOTO 75                                            INT16260
      IM      = I                                                        INT16270
      UM      = UE(IM)                                                   INT16280
75    CONTINUE                                                           INT16290
      IF(IM.LT.4) IM = 4                                                 INT16300
      NTR  = IM                                                          INT16310
      XCTR = XC(NTR)                                                     INT16320
      GOTO 200                                                           INT16330
C                                                                        INT16340
C   TRANSTION LOCATION WILL BE CALCULATED BASED ON MICHEL CRITERION      INT16350
C                                                                        INT16360
150   NTR     = NXT + 1                                                  INT16370
200   DO 210 I=1,NXT                                                     INT16380
210   GMTRS(I)= 0.0                                                      INT16390
C                                                                        INT16400
C  TRANSITION LOCATION PROVISIONALLY FROM PREVIOUS CYCLE                 INT16410
C                                                                        INT16420
      IF (TRFIND(ISF) ) THEN                                             INT16430
         DO 211 I = 1 , NXT                                              INT16440
         XCS(I) = XC(I)                                                  INT16450
         IF (I .LT. MIN) XCS(I) = -XCS(I)                                INT16460
211      CONTINUE                                                        INT16470
         DO 215 I=1,NXT                                                  INT16480
         IF(XCTRS(ISF) .LE.XCS(I)) GOTO 217                              INT16490
215      CONTINUE                                                        INT16500
217      NTR = I-1                                                       INT16510
         XCTR = XCTRS(ISF)                                               INT16520
         IF (NTR .LT. 3) THEN                                            INT16530
              NTR = 3                                                    INT16540
              XCTR = XC(NTR)                                             INT16550
         END IF                                                          INT16560
      END IF                                                             INT16570
C                                                                        INT16580
C  CALCULATE GAMTR DISTRIBUTION                                          INT16590
C                                                                        INT16600
      IF (NTR.GT.NXT-1) GOTO 250                                         INT16610
      FAC  = (XCTR-XC(NTR))/(XC(NTR+1)-XC(NTR))                          INT16620
      XTR  = X(NTR) + FAC*(X(NTR+1)-X(NTR))                              INT16630
      UETR = UE(NTR) + FAC*(UE(NTR+1)-UE(NTR))                           INT16640
      RXNTR  = XTR * UETR * RL                                           INT16650
      GGFT   = 1.0/PGAMTR*RL**2/RXNTR**1.34                              INT16660
      DO 220 I=NTR+1,NXT                                                 INT16670
```

```
      ALFAS(I) = 0.0168                                                  INT16680
220   GMTRS(I)= 1.0                                                      INT16690
      ALFAS(NTR) = 0.0168                                                INT16700
      UEINTG = 0.0                                                       INT16710
      U1 = 0.5/UETR                                                      INT16720
      X1 = XTR                                                           INT16730
      DO 230 I=NTR+1,NXT                                                 INT16740
      U2 = 0.5/UE(I)                                                     INT16750
      X2 = X(I)                                                          INT16760
      UEINTG = UEINTG+(U1+U2)*(X2-X1)                                    INT16770
      U1 = U2                                                            INT16780
      X1 = X2                                                            INT16790
      GG = GGFT*UEINTG*(X(I)-XTR)*UE(I)**3                               INT16800
      IF(GG .GT. 10.0) GO TO 250                                         INT16810
      GMTRS(I) = 1.0-EXP(-GG)                                            INT16820
230   CONTINUE                                                           INT16830
250   CONTINUE                                                           INT16840
C                                                                        INT16850
C   GENERATE B. L. ETA GRIDS + INTIAL VELOCITY PROFILES                  INT16860
C                                                                        INT16870
      CALL INTL(ETAE,DETA1,VGP)                                          INT16880
      DO 260 I=1,NXT                                                     INT16890
      UE0(I)  = UE(I)                                                    INT16900
260   CONTINUE                                                           INT16910
C   PRINT OUT INPUT DATA                                                 INT16920
C                                                                        INT16930
      IF (ICYCLE .GE. ICYTL-1 .OR. IP .GE. 0) THEN                       INT16940
         CALL HEADER( TITLE,SURFID,ISTRP )                               INT16950
         WRITE(6,1002) NXT,ITR,IP,NS,NTR,ISWPT                           INT16960
         WRITE(6,1003) VGP,DETA1,RL,XCTR,OMEGA,PGAMTR                    INT16970
      ELSE                                                               INT16980
         IF (ISF.EQ.1) WRITE(6,1008) ICYCLE                              INT16990
         IF (ISF.EQ.2) WRITE(6,1009) ICYCLE                              INT17000
      END IF                                                             INT17010
      IF (NTR.LT.NXT) THEN                                               INT17020
         IF (ITR.EQ.1) WRITE (6,1005) XCTR,XTR,NTR                       INT17030
         IF (ITR.EQ.2) WRITE (6,1006) XCTR,XTR,NTR                       INT17040
         IF (TRFIND(ISF)) WRITE(6,1007) XCTR,XTR,NTR                     INT17050
      END IF                                                             INT17060
      RETURN                                                             INT17070
C                                                                        INT17080
C   ---------- ----------------------------------------------------      INT17090
C                                                                        INT17100
2     FORMAT(20A4)                                                       INT17110
3     FORMAT(10I5)                                                       INT17120
4     FORMAT(6F10.0)                                                     INT17130
1001  FORMAT(1H1,20X,20A4)                                               INT17140
1002  FORMAT(1H0,10H   NXT  = ,I5,7X,10H   ITR  = ,I5,7X/                INT17150
     +           1H ,10H   IP   = ,I5,7X,10H   NS   = ,I5,7X/           INT17160
     +           1H ,10H   NTR  = ,I5,7X,10H   ISWPT= ,I5)              INT17170
1003  FORMAT(1H0,10H   VGP  = ,E12.4,10H   DETA1= ,E12.4/                INT17180
     +           1H ,10H   RL   = ,E12.4,10H   XCTR = ,E12.4/           INT17190
     +           1H ,10H  OMEGA = ,E12.4,10H  PGAMTR= ,E12.4)           INT17200
1004  FORMAT(1H0,3X,2H I,6X,2HXC,11X,2HYC,11X,2H X,11X,2HUE,11X,2HP1,    INT17210
     +           11X,2HP2,/(1H ,3X,I3,6E13.5))                          INT17220
1005  FORMAT(/3X,'BEGIN OF TRANSITION IS BEING INPUT AT X/C =',F8.4,4X,  INT17230
```

```
     +       'S/C =',F8.4,4X,'NTR =',I3/)                                    INT17240
1006  FORMAT(/3X,'BEGIN OF TRANSITION IS SET AT PRESSURE PEAK, X/C =',       INT17250
     +       F8.4,4X,'S/C =',F8.4,4X,'NTR =',I3/)                            INT17260
1007  FORMAT(/3X,'BEGIN OF TRANSITION IS PROVISIONALLY TAKEN FROM ',         INT17270
     +  'PREVIOUS CYCLE: X/C=',F8.4,4X,'S/C =',F8.4,4X,'NTR =',I3/)         INT17280
1008  FORMAT(//3X,'UPPER SURFACE CALCULATIONS OF CYCLE',I3)                  INT17290
1009  FORMAT(//3X,'LOWER SURFACE CALCULATIONS OF CYCLE',I3)                  INT17300
      END                                                                    INT17310
C          DATA SET KCBCINTL   AT LEVEL 001 AS OF 08/24/84                  INT17320
C          DATA SET KCBCINTL   AT LEVEL 001 AS OF 08/24/84                  INT17330
C          DATA SET KCBCINTL   AT LEVEL 009 AS OF 02/22/84                  INT17340
      SUBROUTINE INTL(ETAE,DETA1,VGP)                                        INT17350
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                         INT17360
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)             INT17370
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)                  INT17380
      COMMON /BLC6/ S1(101),S2(101),S3(101),S4(101),S5(101),S6(101),         INT17390
     +                    S7(101),S8(101),R1(101),R2(101),R3(101),R4(101)INT17400
      COMMON /BONV/ ITMAX,EPSL,EPST,CONV                                     INT17410
      COMMON /GRD / ETA(101),DETA(101),A(101)                                INT17420
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH                  INT17430
C                                                                            INT17440
C     ------------------------------------------------------------           INT17450
C                                                                            INT17460
C    GENERATE THE GRID                                                       INT17470
C                                                                            INT17480
      DETA(1) = DETA1                                                        INT17490
      IF(VGP.LT.1.0) VGP = 1.0                                               INT17500
      IF((VGP-1.0) .LE. 0.001)  GO TO 10                                     INT17510
      NP     = ALOG((ETAE/DETA(1))*(VGP-1.0)+1.0)/ALOG(VGP) + 1.001          INT17520
      GO TO 20                                                               INT17530
10    NP     = ETAE/DETA(1) + 1.001                                          INT17540
20    IF(NP .LE. NPT)  GO TO 30                                              INT17550
      WRITE(6, 150 )                                                         INT17560
      STOP                                                                   INT17570
30    ETA(1) = 0.0                                                           INT17580
      DO 40 J=2,NPT                                                          INT17590
      ETA(J) = ETA(J-1) + DETA(J-1)                                          INT17600
      DETA(J)= VGP*DETA(J-1)                                                 INT17610
      A(J)   = 0.5*DETA(J-1)                                                 INT17620
40    CONTINUE                                                               INT17630
C                                                                            INT17640
C   GENERATE INITIAL VELOCITY PROFILE                                        INT17650
80    DO 90  J=1,NP                                                          INT17660
      ETAB = ETA(J)/ETA(NP)                                                  INT17670
      ETAB2 = ETAB**2                                                        INT17680
      F(J,2) = 0.25*ETA(NP)*ETAB2*(3.0 - 0.5*ETAB2)                          INT17690
      U(J,2) = 0.5*ETAB*(3.0 - ETAB2)                                        INT17700
      V(J,2) = 1.5*(1.0 - ETAB2)/ETA(NP)                                     INT17710
      B(J,2) = 1.0                                                           INT17720
      W(J,2) = 1.0                                                           INT17730
90    CONTINUE                                                               INT17740
      NX     = 1                                                             INT17750
      IT     = 0                                                             INT17760
120   IT     = IT + 1                                                        INT17770
      IF(IT .LT. ITMAX) GO TO 130                                            INT17780
```

```
      STOP                                                              INT17790
130   CONTINUE                                                          INT17800
C                                                                       INT17810
      DO 140 J= 2,NP                                                    INT17820
      FB     = 0.5*(F(J,2) + F(J-1,2))                                  INT17830
      UB     = 0.5*(U(J,2) + U(J-1,2))                                  INT17840
      VB     = 0.5*(V(J,2) + V(J-1,2))                                  INT17850
      USB    = 0.5*(U(J,2)**2 + U(J-1,2)**2)                            INT17860
      DERBV  = (B(J,2)*V(J,2) - B(J-1,2)*V(J-1,2))/DETA(J-1)            INT17870
      FVB    = 0.5*(F(J,2)*V(J,2) + F(J-1,2)*V(J-1,2))                  INT17880
C                                                                       INT17890
      S1(J)  =  0.5*P1(NX)*F(J,2) + B(J,2)/DETA(J-1)                    INT17900
      S2(J)  =  0.5*P1(NX)*F(J-1,2) - B(J-1,2)/DETA(J-1)                INT17910
      S3(J)  = 0.5*P1(NX)*V(J,2)                                        INT17920
      S4(J)  = 0.5*P1(NX)*V(J-1,2)                                      INT17930
      S5(J)  = -P2(NX)*U(J,2)                                           INT17940
      S6(J)  = -P2(NX)*U(J-1,2)                                         INT17950
      CRB    = -P2(NX)                                                  INT17960
      R2(J)  = CRB - (DERBV + P1(NX)*FVB - P2(NX)*USB)                  INT17970
C                                                                       INT17980
      R1(J)  = F(J-1,2) - F(J,2) + DETA(J-1)*UB                         INT17990
      R3(J-1)= U(J-1,2) - U(J,2) + DETA(J-1)*VB                         INT18000
140   CONTINUE                                                          INT18010
      R1(1)  = 0.0                                                      INT18020
      R2(1)  = 0.0                                                      INT18030
      R3(NP) = 0.0                                                      INT18040
      CALL SOLV3                                                        INT18050
      IF(ABS(DELV(1)) .GT. EPSL  ) GO TO 120                            INT18060
      CALL FILLUP(2)                                                    INT18070
      CALL OUTPUT(1)                                                    INT18080
C                                                                       INT18090
      RETURN                                                            INT18100
C                                                                       INT18110
150   FORMAT(1H0,37HNP EXCEEDED NPT -- PROGRAM TERMINATED)              INT18120
      END                                                               INT18130
C          DATA SET KCBCINTRP3 AT LEVEL 001 AS OF 08/24/84              INT18140
C          DATA SET KCBCINTRP3 AT LEVEL 001 AS OF 08/24/84              INT18150
C          DATA SET KCBCINTRP3 AT LEVEL 003 AS OF 04/05/84              INT18160
      SUBROUTINE INTRP3 (N1,X1,F1,FP1,FPP1,FPPP1,N2,X2,F2)              INT18170
C                                                                       INT18180
C     CUBIC INTERPOLATION                                               INT18190
C                                                                       INT18200
C     GIVEN THE VALUES OF A FUNCTION (F1) AND ITS DERIVATIVES           INT18210
C     AT N1 VALUES OF THE INDEPENDENT VARIABLE (X1)                     INT18220
C                                                                       INT18230
C     FIND THE VALUES OF THE FUNCTION (F2)                              INT18240
C     AT N2 VALUES OF THE INDEPENDENT VARIABLE (X2)                     INT18250
C                                                                       INT18260
C     X2 CAN BE IN ARBITRARY ORDER                                      INT18270
C                                                                       INT18280
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   INT18290
C                                                                       INT18300
      DIMENSION X1(101),F1(101),FP1(101),FPP1(101),FPPP1(101),X2(101),  INT18310
     +             F2(101)                                              INT18320
      DATA EPS /1E-04/                                                  INT18330
C                                                                       INT18340
```

```
      JT = 2                                                            INT18350
      DO 10 I = 1,N2                                                    INT18360
      XM      = X2(I)                                                   INT18370
      DO 20 J = JT,N1                                                   INT18380
      J1 = J -1                                                         INT18390
      IF (X1(J).GE.XM ) GO TO 30                                        INT18400
20    CONTINUE                                                          INT18410
      J = N1                                                            INT18420
30    JT = J                                                            INT18430
      DXX     = X2(I) - X1(J1)                                          INT18440
      DXX2    = DXX  * DXX / 2.                                         INT18450
      DXX3    = DXX2 * DXX / 3.                                         INT18460
10    F2(I)   = F1(J1) + DXX*FP1(J1) + DXX2*FPP1(J1) + DXX3*FPPP1(J1)   INT18470
C                                                                       INT18480
      RETURN                                                            INT18490
      END                                                               INT18500
C          DATA SET KCBCJOIN   AT LEVEL 001 AS OF 08/24/84              INT18510
C          DATA SET KCBCJOIN   AT LEVEL 001 AS OF 08/24/84              INT18520
C          DATA SET KCBCJOIN   AT LEVEL 012 AS OF 02/20/84              INT18530
      SUBROUTINE JOIN(INDEX)                                            INT18540
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT18550
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT18560
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI      INT18570
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                    INT18580
     +                    ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT      INT18590
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT18600
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT18610
      COMMON /SAVE/ FS(101),US(101),VS(101),BS(101),WS(101)             INT18620
      COMMON /SMRY/ VW(100),ITP(100),ISL(100),DLS(100),CF(100),         INT18630
     +              THT(100),NPSTR(100)                                 INT18640
C                                                                       INT18650
C   ------------------------------------------------------------        INT18660
C                                                                       INT18670
C  INDEX = 1 FOR THE FIRST SWEEP                                        INT18680
C  INDEX = 2 FOR SUBSEQUENT SWEEP                                       INT18690
C                                                                       INT18700
      CALL COMPGI                                                       INT18710
      CII     = C(NX,NX)                                                INT18720
      UES     = GI / (1.0 - DLS(NX) * SQRT(RL) * CII )                  INT18730
      IF(INDEX.EQ.1) GOTO 15                                            INT18740
C                                                                       INT18750
C   RETRIEVE PROFILES AT STATION NS FOR INVERSE B. L.                   INT18760
C   CALCULATION                                                         INT18770
      DO 10 J=1,NPT                                                     INT18780
      F(J,2) = FS(J)                                                    INT18790
      U(J,2) = US(J)                                                    INT18800
      V(J,2) = VS(J)                                                    INT18810
      W(J,2) = WS(J)                                                    INT18820
      B(J,2) = BS(J)                                                    INT18830
10    CONTINUE                                                          INT18840
      UES     = UES/W(1,2)                                              INT18850
      SQS     = 1.0                                                     INT18860
      GOTO 30                                                           INT18870
15    CONTINUE                                                          INT18880
      SQS     = 1.0 / SQRT(UES)                                         INT18890
      DO 20 J=2,NPT                                                     INT18900
```

```
      ETA(J) = ETA(J)*SQS                                              INT18910
      DETA(J-1) = ETA(J) - ETA(J-1)                                    INT18920
      A(J)   = 0.5*DETA(J-1)                                           INT18930
20    CONTINUE                                                         INT18940
C                                                                      INT18950
30    DO 60 J=1,NPT                                                    INT18960
      U(J,2) = U(J,2)*UES                                              INT18970
      W(J,2) = UES * W(J,2)                                            INT18980
      F(J,2) = F(J,2)*SQS*UES                                          INT18990
      V(J,2) = V(J,2)/SQS*UES                                          INT19000
60    CONTINUE                                                         INT19010
      UE(NX) = W(1,2)                                                  INT19020
      RX     = UE(NX)*X(NX)*RL                                         INT19030
      SQRX   = SQRT(RX)                                                INT19040
C     IF(NX.GT.NTR) CALL EDDY                                          INT19050
      CALL FILLUP(2)                                                   INT19060
      IF(INDEX.EQ.2) GOTO 70                                           INT19070
C                                                                      INT19080
C   STORE PROFILES AT STATION NS FOR THE NEXT SWEEP                    INT19090
      DO 65 J=1,NPT                                                    INT19100
      FS(J)  = F(J,2)                                                  INT19110
      US(J)  = U(J,2)                                                  INT19120
      VS(J)  = V(J,2)                                                  INT19130
      WS(J)  = W(J,2)                                                  INT19140
      BS(J)  = B(J,2)                                                  INT19150
65    CONTINUE                                                         INT19160
70    DO 80 J=1,NPT                                                    INT19170
      F(J,1) = F(J,2)                                                  INT19180
      U(J,1) = U(J,2)                                                  INT19190
      V(J,1) = V(J,2)                                                  INT19200
      W(J,1) = W(J,2)                                                  INT19210
      B(J,1) = B(J,2)                                                  INT19220
80    CONTINUE                                                         INT19230
      RETURN                                                           INT19240
      END                                                              INT19250
C          DATA SET KCBCMAIN   AT LEVEL 005 AS OF 09/18/84             INT19260
C                                                                      INT19270
C      PROGRAM MAIN                                                    INT19280
C                                                                      INT19290
C------------------------------------------------------               INT19300
      SUBROUTINE CASBLP(K2,XP,YP,XMP,YMP,XS,YS,DLSP,VC,DBPP            INT19310
     +                  ,RN,NBL,ITRI,XCTRI,CASEID)                     INT19320
      COMMON /BLCO/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                   INT19330
      COMMON/BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),       INT19340
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)      INT19350
      COMMON/EDDY1/RL,RX,SQRX,RXNTR,GMTR,GMTRS(100),                   INT19360
     +                  ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT        INT19370
      COMMON/BLOW/ VN(100)                                             INT19380
      COMMON/TRN/ PGAMTR,OMEGA,RTHETB,RTRANB                           INT19390
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                INT19400
      DIMENSION     CASEID( 20 ),  XCTRI (  2 ),  ITRI  (  2 )         INT19410
      DIMENSION     XP    ( 100),  YP    ( 100),  XMP   ( 100)        INT19420
      DIMENSION     YMP   ( 100),  VC    ( 100),  SM    ( 100)        INT19430
      DIMENSION     XS    ( 100),  YS    ( 100),  NBL   ( 2 )         INT19440
      DIMENSION     VT    ( 100),  S     ( 100),  DLSP  ( 100)        INT19450
      DIMENSION     DLS   ( 100),  XO    ( 100),  YO    ( 100)        INT19460
```

```
      DIMENSION     D1    ( 100),  D2    ( 100),  D3    ( 100)          INT19470
      DIMENSION     XPS   (100),  YPS(100) , DBPP(100)                  INT19480
C                                                                       INT19490
10    FORMAT ( 20A4 )                                                   INT19500
20    FORMAT ( 4I5 )                                                    INT19510
30    FORMAT ( 3I5,3F10.0 )                                             INT19520
40    FORMAT ( 6F10.5 )                                                 INT19530
100   FORMAT ( 1H1,5X,20A4 )                                            INT19540
110   FORMAT ( 1H0,6X,'CYCLE NO. = ',I5 )                               INT19550
120   FORMAT ( 1H0,6X,'FINISHED TOTAL NUMBER OF CYCLES = ',I5,          INT19560
     +              4X,'JOB COMPLETED.' )                              INT19570
130   FORMAT ( 1H0,6X,'THE UPDATED DISPLACEMENT SURFACE',/1X,2HNX,      INT19580
     +              9X,2HXP,12X,2HYP,11X,3HDLS,10X,4HDBPP,12X,2HVN)    INT19590
140   FORMAT ( I5,5E14.6)                                               INT19600
150   FORMAT ( 1H0.6X,'READ IN CONTROL POINTS DATA',/1X,2HNX,9X,        INT19610
     +                3HXMP,11X,3HYMP,12X,2HSM,13X,2HVC)               INT19620
160   FORMAT ( I5,4E14.6)                                               INT19630
170   FORMAT ( 1H0,6X,'INTERPOLATED ORIGINAL GEOMETRY DATA',/1X,        INT19640
     +                2HNX,9X,2HXP,12X,2HYP,12X,1HS,14X,2HVT)           INT19650
180   FORMAT (I5,4E14.6)                                                INT19660
C                                                                       INT19670
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -     INT19680
C                                                                       INT19690
      GRANG(X1,X2,X3,Y1,Y2,Y3,X0)= (X0-X2)*(X0-X3)/(X1-X2)/(X1-X3)*Y1   INT19700
     +          +(X0-X1)*(X0-X3)/(X2-X1)/(X2-X3)*Y2+(X0-X1)*(X0-X2)    INT19710
     +          /(X3-X1)/(X3-X2)*Y3                                     INT19720
      ISWPT = 1                                                         INT19730
      IEDY  = 1                                                         INT19740
      NBL(1) = 91                                                       INT19750
      NBL(2) = 71                                                       INT19760
C                                                                       INT19770
C     WRITE ( 6,100 ) CASEID                                            INT19780
5     CONTINUE                                                          INT19790
      ISTRP = ICYCLE                                                    INT19800
      NN = K2 - 1                                                       INT19810
      NT = K2                                                           INT19820
C      WRITE ( 6,110 ) ICYCLE                                           INT19830
C  INTERPOLATE OUTPUT CONTROL POINTS TO ORIGINAL GEOMETRY               INT19840
C                                                                       INT19850
      DO 15 I = 1 , NT                                                  INT19860
      XPS(I) = XP(I)                                                    INT19870
      YPS(I) = YP(I)                                                    INT19880
15    CONTINUE                                                          INT19890
      S(1) = 0.0                                                        INT19900
      DO 50 I = 2 , NT                                                  INT19910
      S(I) = S(I-1) + SQRT((XP(I)-XP(I-1))**2 +                         INT19920
     +                          (YP(I)-YP(I-1))**2)                     INT19930
50    CONTINUE                                                          INT19940
      SM(1) = SQRT((XMP(1)-XP(1))**2 + (YMP(1)-YP(1))**2)               INT19950
      DO 60 I = 2 , NN                                                  INT19960
      SM(I) = SM(I-1) + SQRT((XMP(I)-XMP(I-1))**2+                      INT19970
     +                            (YMP(I)-YMP(I-1))**2)                 INT19980
60    CONTINUE                                                          INT19990
C     CALL AMEAN(NN-10,NN,SM,VC,1)                                      INT20000
      CALL AMEAN(1,NN,SM,VC,1)                                          INT20010
      SNT = S(NT)                                                       INT20020
```

```
      SM(NT) = SM(NN)+SQRT((XMP(NN)-XP(NT))**2+(YMP(NN)-YP(NT))**2)     INT20030
      SMNT = S(NT) / SM(NT)                                             INT20040
      DO 65 I = 1 , NN                                                  INT20050
65    SM(I) = SM(I) * SMNT                                              INT20060
      CALL DIFF3(NN,SM,VC,D1,D2,D3,0)                                   INT20070
      CALL INTRP3(NN,SM,VC,D1,D2,D3,NT,S,VT)                            INT20080
C  PRINT OUT INPUT DATA                                                 INT20090
C                                                                       INT20100
C     WRITE(6,150)                                                      INT20110
C     WRITE(6,160) (I,XMP(I),YMP(I),SM(I),VC(I),I=1,NN)                 INT20120
C     WRITE(6,170)                                                      INT20130
C     WRITE(6,180) (I,XP(I),YP(I),S(I),VT(I),I=1,NT)                    INT20140
C                                                                       INT20150
      XPMIN = XP(1)                                                     INT20160
      DO 44 I = 2 , NT                                                  INT20170
      IF (XP(I) .GT. XPMIN) GO TO 44                                    INT20180
      XPMIN = XP(I)                                                     INT20190
44    CONTINUE                                                          INT20200
      CHORD = XP(NT) - XPMIN                                            INT20210
      DO 45 I = 1 , NT                                                  INT20220
      XP(I) = (XP(I)-XPMIN) / CHORD                                     INT20230
      YP(I) = YP(I) / CHORD                                             INT20240
45    CONTINUE                                                          INT20250
      CALL COMPBL ( CASEID,XP,YP,VT,S,DLSP,DLS,DBPP,NBL,ITRI,XCTRI,     INT20260
     +                    RN,NT,ISWPT)                                 INT20270
C                                                                       INT20280
C                                                                       INT20290
C  SMOOTH THE CALCULATED DISPLACEMENT THINKNESS                         INT20300
C                                                                       INT20310
C     CALL SMFIT(1,NT,S,DLS,D1,2)                                       INT20320
C                                                                       INT20330
C  ADD DISPLACEMENT THINKNESS ON THE ORIGINAL BODY                      INT20340
C                                                                       INT20350
      DO 70 I = 1 , NT                                                  INT20360
      DLSP(I) = DLS(I)                                                  INT20370
70    CONTINUE                                                          INT20380
C     CALL SMFIT (1,NT,XS,YS,D1,2)                                      INT20390
      DO 80 I = 1 , NT                                                  INT20400
      XP(I) = XPS(I)                                                    INT20410
      YP(I) = YPS(I)                                                    INT20420
80    CONTINUE                                                          INT20430
      IF (ICYCLE .GE. ICYTL-1 .OR. IP .GE. 0) THEN                      INT20440
C      WRITE (6,130)                                                    INT20450
C      WRITE (6,140) (I,XP(I),YP(I),DLS(I),DBPP(I),VN(I),I=1,NT)        INT20460
      WRITE ( 6,120 ) ICYCLE                                            INT20470
      END IF                                                            INT20480
C                                                                       INT20490
      RETURN                                                            INT20500
      END                                                               INT20510
C          DATA SET KCBCMAIN2  AT LEVEL 001 AS OF 08/24/84              INT20520
C          DATA SET KCBCMAIN2  AT LEVEL 001 AS OF 08/24/84              INT20530
C          DATA SET KCBCMAIN2  AT LEVEL 010 AS OF 04/06/84              INT20540
      SUBROUTINE MAIN2(ITR,ISWPT,SURFID)                                INT20550
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT20560
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT20570
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)             INT20580
```

```
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI      INT20590
      COMMON /BONV/ ITMAX,EPSL,EPST,CONV                                INT20600
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                    INT20610
     +                   ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT      INT20620
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT20630
      COMMON /SMRY/ VW(100),ITP(100),ISL(100),DLS(100),CF(100),         INT20640
     +                    THT(100),NPSTR(100)                           INT20650
      COMMON /BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),       INT20660
     +              XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)      INT20670
      COMMON/BLC9/ UEB(100) , CFS(100)                                  INT20680
      COMMON/TRN/ PGAMTR,OMEGA,RTHETB,RTRANB                            INT20690
      COMMON /ISURF/ ISF                                                INT20700
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                 INT20710
      DIMENSION SURFID(4),RTSS(11)                                      INT20720
      LOGICAL SMOOTH , SEPART , HOMOPY                                  INT20730
      DATA RTSS/0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0/            INT20740
                                                                        INT20750
C     ----------------------------------------------------------------INT20760
C                                                                       INT20770
      GRANG(X1,X2,X3,Y1,Y2,Y3,X0)= (X0-X2)*(X0-X3)/(X1-X2)/(X1-X3)*Y1   INT20780
     +           +(X0-X1)*(X0-X3)/(X2-X1)/(X2-X3)*Y2+(X0-X1)*(X0-X2)    INT20790
     +           /(X3-X1)/(X3-X2)*Y3                                    INT20800
      ISWP   = 0                                                        INT20810
      INDEX  = 1                                                        INT20820
      IGROWT = 2                                                        INT20830
      NXSPT = NXT + 1                                                   INT20840
10    CALL JOIN(INDEX)                                                  INT20850
      NXSTOP = NXT-1                                                    INT20860
      IF (NS .GE. NTR ) GOTO 15                                         INT20870
15    ISWP   = ISWP + 1                                                 INT20880
20    NX     = NX + 1                                                   INT20890
      HOMOPY = .FALSE.                                                  INT20900
25    CEL    = 0.5*(X(NX)+X(NX-1))/(X(NX)-X(NX-1))                      INT20910
      P1(NX) = 0.5                                                      INT20920
      P2(NX) = 0.0                                                      INT20930
      CELH   = 0.5*CEL                                                  INT20940
30    IT     = 0                                                        INT20950
      CALL COMPGI                                                       INT20960
      IGROW=1                                                           INT20970
70    IT     = IT + 1                                                   INT20980
      RX     = UE(NX)*X(NX)*RL                                          INT20990
      SQRX   = SQRT(RX)                                                 INT21000
C                                                                       INT21010
      IF(IT .LE. ITMAX) GO TO 80                                        INT21020
      IF(HOMOPY) GO TO 72                                               INT21030
      IRC = 1                                                           INT21040
      RT  = RTSS(IRC)                                                   INT21050
      HOMOPY = .TRUE.                                                   INT21060
      UEREF  = UE0(NX-1)                                                INT21070
      UESAVE = UE0(NX)                                                  INT21080
      UE0(NX) = RT*UESAVE+(1.0-RT)*UEREF                                INT21090
      DO 61 J=1,NP                                                      INT21100
      F(J,2) = F(J,1)                                                   INT21110
      U(J,2) = U(J,1)                                                   INT21120
      V(J,2) = V(J,1)                                                   INT21130
      W(J,2) = W(J,1)                                                   INT21140
```

```
      B(J,2) = B(J,1)                                                   INT21150
61    CONTINUE                                                          INT21160
      GO TO 30                                                          INT21170
C                                                                       INT21180
72    NXSTOP = NX - 1                                                   INT21190
      CALL AMEAN(NS,NXSTOP,X,CF,1)                                      INT21200
C     CALL AMEAN(NS,NXSTOP,X,VW,1)                                      INT21210
C     CALL HEADER( TITLE,SURFID,ISTRP )                                 INT21220
      WRITE (6, 250 ) ISWP                                              INT21230
      WRITE (6, 260 ) (M,XC(M),X(M),CF(M),DLS(M),THT(M),UE(M),          INT21240
     +    UE0(M),D(M),DB(M),GMTRS(M),ITP(M),NPSTR(M),M=1,NXSTOP)        INT21250
      WRITE(6, 270 ) NX                                                 INT21260
      STOP                                                              INT21270
80    CONTINUE                                                          INT21280
      IF(NX .GT. NTR) GOTO 100                                          INT21290
C                                                                       INT21300
C   LAMINAR FLOW CALCULATION                                            INT21310
C                                                                       INT21320
      CALL COEF(GAMMA1,GAMMA2)                                          INT21330
      CALL SOLV4(GAMMA1,GAMMA2)                                         INT21340
      UE(NX) = U(NP,2)                                                  INT21350
      IF(ABS(DELV(1)) .GT. EPSL) GO TO 70                               INT21360
C                                                                       INT21370
C  CHECK ON LAMINAR FLOW SEPARATION. IF SEPARATION OCCURS, ASSIGN BEGIN INT21380
C  OF TRANSITION TO THAT POINT AND RECOMPUTE THE CURRENT STATION NX     INT21390
C                                                                       INT21400
      IF(V(1,2).GT.0.0 .OR. ITR.NE.3) GOTO 110                          INT21410
      CALL TRNS(ICODE)                                                  INT21420
      GOTO 25                                                           INT21430
C                                                                       INT21440
C   TURBULENT FLOW CALCULATION                                          INT21450
C                                                                       INT21460
100   CONTINUE                                                          INT21470
      CALL EDDY                                                         INT21480
      CALL COEF(GAMMA1,GAMMA2)                                          INT21490
      CALL SOLV4(GAMMA1,GAMMA2)                                         INT21500
      UE(NX) = U(NP,2)                                                  INT21510
      VM     = AMAX1(V(1,2),1.0)                                        INT21520
      IF(ABS(DELV(1)/VM) .GT. EPST) GO TO 70                            INT21530
110   CONTINUE                                                          INT21540
C                                                                       INT21550
C     CHECK FOR B. L. GROWTH                                            INT21560
C                                                                       INT21570
      IF(NP .GE. NPT) GO TO 120                                         INT21580
      IF(ABS(V(NP,2)) .LT. 0.0005 .AND. ABS(1.0-U(NP-2,2)/U(NP,2))      INT21590
     +              .LT.0.0035.OR.IGROW.GT.IGROWT) GOTO 120             INT21600
      CALL FILLUP(1)                                                    INT21610
      IGROW=IGROW+1                                                     INT21620
      IT    = 1                                                         INT21630
      GO TO 70                                                          INT21640
C                                                                       INT21650
120   CONTINUE                                                          INT21660
      CALL FILLUP(2)                                                    INT21670
      CALL OUTPUT(2)                                                    INT21680
      IF(NX.GE.NTR  .OR.  ITR.EQ.0) GOTO 150                            INT21690
```

```
      IF(NX.LT.3 .OR.   ITR.NE.3) GOTO 150                              INT21700
C                                                                       INT21710
C   CALCULATE TRANSITION LOCATION USING MICHEL METHOD                   INT21720
C                                                                       INT21730
      CALL TRNS(ICODE)                                                  INT21740
      IF(ICODE.EQ.0) GOTO 150                                           INT21750
C                                                                       INT21760
C   TRANSITION OCCURS BASED ON MICHEL CRITERIOR AT STATION NX           INT21770
C   RECALCULATE B. L. AT NX STATION ASSUMING THE FLOW IS TRANSITIONAL   INT21780
C                                                                       INT21790
      IT      = 0                                                       INT21800
      IGROW   = 1                                                       INT21810
      GOTO 70                                                           INT21820
150   CONTINUE                                                          INT21830
      IF(.NOT. HOMOPY ) GO TO 154                                       INT21840
      IF( RT .GT. 0.9999) GO TO 154                                     INT21850
      IRC = IRC + 1                                                     INT21860
      RT  = RTSS(IRC)                                                   INT21870
      UE0(NX) =RT*UESAVE + (1.0-RT)*UEREF                               INT21880
      GO TO 30                                                          INT21890
154   CONTINUE                                                          INT21900
      IF(NX .LT. NXSTOP) GO TO 20                                       INT21910
C                                                                       INT21920
C   THE B. L. CALCULATION FOR THE CURRENT SWEEP IS COMPLETED.           INT21930
C   CHECK FOR THE CONVERGENCE AND , IT NOT, MOVE TO THE NEXT            INT21940
C   SWEEP.                                                              INT21950
C                                                                       INT21960
C                                                                       INT21970
160   CONTINUE                                                          INT21980
      D(NXT)  = GRANG(X(NXT-3),X(NXT-2),X(NXT-1),D(NXT-3),D(NXT-2),     INT21990
     +               D(NXT-1),X(NXT))                                   INT22000
      DLS(NXT)= GRANG(X(NXT-3),X(NXT-2),X(NXT-1),DLS(NXT-3),DLS(NXT-2), INT22010
     +               DLS(NXT-1),X(NXT))                                 INT22020
      UE(NXT) = GRANG(X(NXT-3),X(NXT-2),X(NXT-1),UE(NXT-3),             INT22030
     +                    UE(NXT-2),UE(NXT-1),X(NXT))                   INT22040
      DO 165 I = 1 , NXSTOP                                             INT22050
165   CFS(I) = CF(I)                                                    INT22060
      CALL AMEAN(NS,NXSTOP,X,CF,1)                                      INT22070
C     CALL AMEAN(NS,NXSTOP,X,VW,1)                                      INT22080
C     CALL HEADER( TITLE,SURFID,ISTRP )                                 INT22090
      IF(ICYCLE .LT. ICYTL-1 .AND. IP .LT. 0)GO TO 170                  INT22100
      WRITE (6, 250 ) ISWP                                              INT22110
      WRITE (6, 262 ) 1,XC(1),X(1),CF(1),DLS(1),THT(1),UE(1),           INT22120
     +                UE0(1),0.0,GMTRS(1),ITP(1),NPSTR(1)               INT22130
      WRITE (6, 264 ) (M,XC(M),X(M),CF(M),DLS(M),THT(M),UE(M),          INT22140
     +     UE0(M),DLS(M)/THT(M),GMTRS(M),ITP(M),NPSTR(M),M=2,NXSTOP)    INT22150
      IF ((ICYCLE.EQ.ICYTL).AND.(IP.EQ.-2).AND.(NVP(ISF).NE.0)) THEN    INT22160
         WRITE(12,800) NS+1,NTR,XCTR                                    INT22170
         WRITE(12,810) (XC(M),X(M),UE(M),CF(M),GMTRS(M),ISG(M),         INT22180
     2                  M=1,NXSTOP)                                     INT22190
 800     FORMAT(2I5,F10.6)                                              INT22200
 810     FORMAT(5E15.5,I5)                                              INT22210
      END IF                                                            INT22220
170   CONTINUE                                                          INT22230
C                                                                       INT22240
      DMAX   = D(1)                                                     INT22250
```

```
      DDMAX  = ABS(D(1) - DB(1))                                        INT22260
      DO 180 I = 2,NXT                                                  INT22270
      DMAX   = AMAX1( DMAX,D(I) )                                       INT22280
      DD     = ABS(D(I) - DB(I))                                        INT22290
      DDMAX  = AMAX1( DDMAX,DD )                                        INT22300
180   CONTINUE                                                          INT22310
      IF ( ABS( DDMAX / DMAX ) .LE. 0.0050 ) RETURN                     INT22320
C                                                                       INT22330
C                                                                       INT22340
C   UPDATE D FOR THE NEXT SWEEP                                         INT22350
C                                                                       INT22360
      IF (ISWP .GT. 1) GO TO 195                                        INT22370
      DO 190 I = NS , NXT                                               INT22380
190   D(I) = D(I)*(1.0+OMEGA*(UE(I)/UE0(I)-1.0))                        INT22390
      GO TO 205                                                         INT22400
195   IF (ISWP .EQ. 2) GOTO 205                                         INT22410
      DO 200 I = NS , NXT                                               INT22420
200   D(I) = D(I) * (1.0+OMEGA*(UE(I)/UEB(I)-1.0))                      INT22430
205   IF(ISWP .GE. ISWPT ) RETURN                                       INT22440
      NX      = NS                                                      INT22450
      NP      = NPSTR(NX)                                               INT22460
      INDEX   = 2                                                       INT22470
      DO 210 I= 1,NXT                                                   INT22480
      DB(I)   = D(I)                                                    INT22490
      UEB(I) = UE(I)                                                    INT22500
210   CONTINUE                                                          INT22510
      GOTO 10                                                           INT22520
C  ------------------------------------------------------------         INT22530
250   FORMAT(1H0,' ** SUMMARY OF INVERSE BOUNDARY LAYER SOLUTIONS. **',/INT22540
     +           1H0,4X,'ISWP  =',I3/)                                  INT22550
260   FORMAT(1H0,4X,2HNX,5X,3HX/C,9X,1HX,9X,2HCF, 8X,                   INT22560
     +           3HDLS,8X,3HTHT,9X,2HUE, 8X,3HUE0,10X,1HD ,9X,2HDB,3X,  INT22570
     +       4HGMTR,4X,2HIT,1X,2HNP/(1H ,3X,I3,F10.5,8E11.4,F8.4,2I3))  INT22580
262   FORMAT(1H0,4X,2HNX,6X,3HX/C,11X,1HX,10X,2HCF,9X,                  INT22590
     +           3HDLS,9X,3HTHT,10X,2HUE,9X,3HUE0,11X,1HH,8X,           INT22600
     +       4HGMTR,4X,2HIT,1X,2HNP/(1H ,3X,I3,9E12.4,2I3))            INT22610
264   FORMAT(1H ,3X,I3,9E12.4,2I3)                                      INT22620
270   FORMAT(1H0,' ** ITERATIONS EXCEEDED ITMAX AT NX =',I5,',**',/     INT22630
     +           1H0,' ** CALCULATIONS STOP. **')                       INT22640
      END                                                               INT22650
C          DATA SET KCBCOUTPUT AT LEVEL 001 AS OF 08/24/84              INT22660
C          DATA SET KCBCOUTPUT AT LEVEL 001 AS OF 08/24/84              INT22670
C          DATA SET KCBCOUTPUT AT LEVEL 002 AS OF 02/22/84              INT22680
      SUBROUTINE OUTPUT(INDEX)                                          INT22690
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT22700
      COMMON/BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),        INT22710
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)       INT22720
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT22730
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI      INT22740
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                    INT22750
     +                   ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT      INT22760
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT22770
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT22780
      COMMON /SMRY/ VW(100),ITP(100),ISL(100),DLS(100),CF(100),THT(100),INT22790
     +                  NPSTR(100)                                      INT22800
      COMMON /ISURF/ ISF                                                INT22810
```

```
      COMMON/PLOT/NVP(2),NXVP(20,2),ICC                                 INT22820
C                                                                       INT22830
C   ---------------------------------------------------------------     INT22840
C                                                                       INT22850
      ITP(NX) = IT                                                      INT22860
      NPSTR(NX)=NP                                                      INT22870
      IF(NX.GT.1) GOTO 5                                                INT22880
      DLS(NX)= 0.0                                                      INT22890
      VW(NX) = 0.0                                                      INT22900
      D(NX)  = 0.0                                                      INT22910
      THT(NX)= 0.0                                                      INT22920
      CF(NX) = 0.0                                                      INT22930
      VW(NX) = 0.0                                                      INT22940
      GOTO 150                                                          INT22950
5     GOTO (10,100,200), INDEX                                          INT22960
C                                                                       INT22970
C   CALCULATE B. L. PARAMETERS FOR TRANSFORMED COORDINATES              INT22980
10    CONTINUE                                                          INT22990
      CF(NX) = 2.0 * V(1,2) * B(1,2)/SQRX                               INT23000
      VW(NX) = UE(NX)*SQRT(UE(NX)/X(NX))*V(1,2)                         INT23010
      DLS(NX)= X(NX)/SQRX * (ETA(NP)-F(NP,2))                           INT23020
      D(NX)  = UE(NX) * DLS(NX) * SQRT(RL)                              INT23030
      U1     = U(1,2) * (1.0 -U(1,2))                                   INT23040
      SUM    = 0.0                                                      INT23050
      DO 20 J=2,NP                                                      INT23060
      U2     = U(J,2) * (1.0 -U(J,2))                                   INT23070
      SUM    = SUM + A(J) * (U1 + U2)                                   INT23080
      U1     = U2                                                       INT23090
20    CONTINUE                                                          INT23100
      THT(NX)= X(NX)/SQRX * SUM                                         INT23110
      GOTO 150                                                          INT23120
C                                                                       INT23130
C   CALCULATE B. L. PARAMETERS FOR SEMI-TRANSF COORDINATES              INT23140
100   CONTINUE                                                          INT23150
      SQXC = SQRT(X(NX))                                                INT23160
      SQRL = SQRT(RL)                                                   INT23170
      CF(NX) = 2.0 * V(1,2) * B(1,2)/(SQXC*SQRL*W(NP,2)**2)             INT23180
      VW(NX) = V(1,2) / SQXC                                            INT23190
      UE(NX) = U(NP,2)                                                  INT23200
      RX = RL * UE(NX) * X(NX)                                          INT23210
      DLS(NX) = (ETA(NP)-F(NP,2)/U(NP,2))/SQRL*SQXC                     INT23220
      SUM     = 0.0                                                     INT23230
      U1      = U(1,2)/U(NP,2)*(1.0 -U(1,2)/U(NP,2))                    INT23240
      DO 120 J=2,NP                                                     INT23250
      U2      = U(J,2)/U(NP,2)*(1.0 -U(J,2)/U(NP,2))                    INT23260
      SUM     = SUM + A(J) * (U1 + U2)                                  INT23270
      U1      = U2                                                      INT23280
120   CONTINUE                                                          INT23290
      THT(NX) = SUM /SQRL * SQXC                                        INT23300
      D(NX)   = (U(NP,2)*ETA(NP)-F(NP,2)) * SQXC                        INT23310
150   IF (NX .GE. NXT) GO TO 160                                        INT23320
      IF (IEDY .EQ. 0 .OR. NX .LE. NTR+2) GO TO 160                     INT23330
C                                                                       INT23340
C  MODIFY ALFA USING SIMPSON'S ARGUMENTS                                INT23350
C                                                                       INT23360
```

```
      CALL SMPSON                                                       INT23370
160   DO 175 J=1,NPT                                                    INT23380
      F(J,1)  = F(J,2)                                                  INT23390
      U(J,1)  = U(J 2)                                                  INT23400
      V(J,1)  = V(J,2)                                                  INT23410
      W(J,1)  = W(J,2)                                                  INT23420
      B(J,1)  = B(J.2)                                                  INT23430
175   CONTINUE                                                          INT23440
      IF ((IP.LE.0).AND.((IP.NE.-2).OR.(ICYCLE.LT.ICYTL))) RETURN       INT23450
C                                                                       INT23460
C   PRINT OUT VELOCITY PROFILES                                         INT23470
200   IF (NX.EQ.1)  GOTO 210                                            INT23480
      IF (NX.LE.NS) THEN                                                INT23490
         FAC1 = SQRT(X(NX)/RL/UE(NX))                                   INT23500
         FAC2 = 1.0                                                     INT23510
      ELSE                                                              INT23520
         FAC1 = SQRT(X(NX)/RL)                                          INT23530
         FAC2 = 1.0/UE(NX)                                              INT23540
      ENDIF                                                             INT23550
      NPM1   = NP -1                                                    INT23560
      WRITE(6,4001) NX,X(NX)                                            INT23570
      WRITE(6,4000)                                                     INT23580
      WRITE(6,4100) (J,ETA(J),F(J,2),U(J,2),V(J,2),W(J,2),B(J,2),       INT23590
     +               ETA(J)*FAC1,U(J,2)*FAC2,J=1,NPM1,3)                INT23600
      WRITE(6,4100) NP,ETA(NP),F(NP,2),U(NP,2),V(NP,2),W(NP,2),B(NP,2), INT23610
     +              ETA(NP)*FAC1,U(NP,2)*FAC2                           INT23620
C                                                                       INT23630
210   IF (IP.NE.-2) RETURN                                              INT23640
      IF ((NXVP(ICC,ISF).NE.NX).OR.(ICC.GT.NVP(ISF))) RETURN            INT23650
      WRITE(12,4200) NP                                                 INT23660
      WRITE(12,4300) (ETA(J),J=1,NP)                                    INT23670
      WRITE(12,4300) (U(J,2),J=1,NP)                                    INT23680
      ICC = ICC+1                                                       INT23690
      RETURN                                                            INT23700
4001  FORMAT(/1H0,'NX =',I5,'  S/C =',F10.5)                            INT23710
4000  FORMAT(1H0,2H J,9X,3HETA,15X,1HF,13X,1HU,13X,1HV,13X,1HW,13X,1HB, INT23720
     +       13X,3HY/C,10X,4HU/UE)                                      INT23730
4100  FORMAT(1H ,I3,E14.5,2X,5E14.5,2X,2E14.5)                          INT23740
4200  FORMAT(I5)                                                        INT23750
4300  FORMAT(8F10.6)                                                    INT23760
      END                                                               INT23770
C          DATA SET KCBCSMFIT  AT LEVEL 001 AS OF 08/24/84              INT23780
C          DATA SET KCBCSMFIT  AT LEVEL 001 AS OF 08/24/84              INT23790
C          DATA SET KCBCSMFIT  AT LEVEL 001 AS OF 08/15/83              INT23800
      SUBROUTINE SMFIT(NS,ND,X,Q,D,KS)                                  INT23810
C                                                                       INT23820
C   THIS SUBROUTINE SMOOTHES DATA, Q, USING FIVE-POINT FORMULA.         INT23830
C                                                                       INT23840
C   NS  : BEGINNING POINT?   ND : END POINT                             INT23850
C   X   : INDEDENPENT COORDINATE? D  : WORKING STORAGE                  INT23860
C   Q   : VARIABLE TO BE SMOOTHED                                       INT23870
C   KS  : NO OF SMOOTHING                                               INT23880
      DIMENSION X(101),Q(101),D(101)                                    INT23890
C   ----------------------------------------------------------------    INT23900
      SMT5(Q1,Q2,Q3,Q4,Q5    ) = 0.0625*(10.0*Q3+4.0*(Q2+Q4)-Q1-Q5)      INT23910
      SMT3(Q1,Q2,Q3,X1,X2,X3) = 0.5*(Q2+(Q1*ABS(X3-X2)+Q3*ABS(X2-X1))   INT23920
```

```
     +                                  /ABS(X3-X1))                          INT23930
C                                                                             INT23940
      IF(KS.LE.0) RETURN                                                      INT23950
      NSP1   = NS+1                                                           INT23960
      NSP2   = NS+2                                                           INT23970
      NDM1   = ND-1                                                           INT23980
      NDM2   = ND-2                                                           INT23990
C                                                                             INT24000
      NDIF   = ND-NS+1                                                        INT24010
      IF ( NDIF .LT. 3 ) RETURN                                               INT24020
      IF ( NDIF .LT. 5 ) GO TO 200                                            INT24030
C                                                                             INT24040
      DO 100 K=1,KS                                                           INT24050
      D(NS+1)= SMT3(Q(NS),Q(NS+1),Q(NS+2),X(NS),X(NS+1),X(NS+2))              INT24060
      D(ND-1)= SMT3(Q(ND-2),Q(ND-1),Q(ND),X(ND-2),X(ND-1),X(ND))              INT24070
      DO 20 I=NSP2,NDM2                                                       INT24080
      D(I)   = SMT5(Q(I-2),Q(I-1),Q(I),Q(I+1),Q(I+2))                         INT24090
20    CONTINUE                                                                INT24100
      DO 40 I=NSP1,NDM1                                                       INT24110
40    Q(I)    = D(I)                                                          INT24120
100   CONTINUE                                                                INT24130
      RETURN                                                                  INT24140
C                                                                             INT24150
200   DO 300 K = 1,KS                                                         INT24160
      DO 220 I = NSP1,NDM1                                                    INT24170
      D(I)   = SMT3(Q(I-1),Q(I),Q(I+1),X(I-1),X(I),X(I+1))                    INT24180
220   CONTINUE                                                                INT24190
      DO 250 I = NSP1,NDM1                                                    INT24200
      Q(I)   = D(I)                                                           INT24210
250   CONTINUE                                                                INT24220
300   CONTINUE                                                                INT24230
C                                                                             INT24240
      RETURN                                                                  INT24250
      END                                                                     INT24260
C          DATA SET KCBCSOLV3  AT LEVEL 001 AS OF 08/24/84                    INT24270
C          DATA SET KCBCSOLV3  AT LEVEL 001 AS OF 08/24/84                    INT24280
C          DATA SET KCBCSOLV3  AT LEVEL 005 AS OF 02/21/84                    INT24290
      SUBROUTINE SOLV3                                                        INT24300
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                          INT24310
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)              INT24320
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)                   INT24330
      COMMON /BLC6/ S1(101),S2(101),S3(101),S4(101),S5(101),S6(101),          INT24340
     +                   S7(101),S8(101),R1(101),R2(101),R3(101),R4(101)INT24350
      COMMON /GRD / ETA(101),DETA(101),A(101)                                 INT24360
      COMMON /BLCB/ A11(101),A12(101),A13(101),A14(101),                      INT24370
     +                   A21(101),A22(101),A23(101),A24(101)                  INT24380
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT24390
      A11(1)= 1.0                                                             INT24400
      A12(1)= 0.0                                                             INT24410
      A13(1)= 0.0                                                             INT24420
      A21(1)= 0.0                                                             INT24430
      A22(1)= 1.0                                                             INT24440
      A23(1)= 0.0                                                             INT24450
      G11=-1.0                                                                INT24460
      G12=-A(2)                                                               INT24470
      G13= 0.0                                                                INT24480
```

```
      G21= S4(2)                                                        INT24490
      G23=-S2(2)/A(2)                                                   INT24500
      G22= G23+S6(2)                                                    INT24510
C                                                                       INT24520
C  FORWARD SWEEP                                                        INT24530
C                                                                       INT24540
      DO 500 J=2,NP                                                     INT24550
      IF(J .EQ. 2) GO TO 100                                            INT24560
      DEN   = (A13(J-1)*A21(J-1)-A23(J-1)*A11(J-1)-A(J)*                INT24570
     +             (A12(J-1)*A21(J-1)-A22(J-1)*A11(J-1)))               INT24580
      DEN1  = A22(J-1)*A(J)-A23(J-1)                                    INT24590
      G11= (A23(J-1)+A(J)*(A(J)*A21(J-1)-A22(J-1)))/DEN                 INT24600
      G12= -(A(J)*A(J)+G11*(A12(J-1)*A(J)-A13(J-1)))/DEN1               INT24610
      G13= (G11*A13(J-1)+G12*A23(J-1))/A(J)                             INT24620
      G21= (S2(J)*A21(J-1)-S4(J)*A23(J-1)+A(J)*(S4(J)*                  INT24630
     +             A22(J-1)-S6(J)*A21(J-1)))/DEN                        INT24640
      G22= (-S2(J)+S6(J)*A(J)-G21*(A(J)*A12(J-1)-A13(J-1)))/DEN1        INT24650
      G23= G21*A12(J-1)+G22*A22(J-1)-S6(J)                              INT24660
100   A11(J)= 1.0                                                       INT24670
      A12(J)=-A(J)-G13                                                  INT24680
      A13(J)= A(J)*G13                                                  INT24690
      A21(J)= S3(J)                                                     INT24700
      A22(J)= S5(J)-G23                                                 INT24710
      A23(J)= S1(J)+A(J)*G23                                            INT24720
      R1(J) = R1(J)-(G11*R1(J-1)+G12*R2(J-1)+G13*R3(J-1))               INT24730
      R2(J) = R2(J)-(G21*R1(J-1)+G22*R2(J-1)+G23*R3(J-1))               INT24740
500   CONTINUE                                                          INT24750
C                                                                       INT24760
C  BACKWARD SWEEP                                                       INT24770
C                                                                       INT24780
      DELU(NP) = R3(NP)                                                 INT24790
      E1       = R1(NP)-A12(NP)*DELU(NP)                                INT24800
      E2       = R2(NP)-A22(NP)*DELU(NP)                                INT24810
      DELV(NP) = (E2*A11(NP)-E1*A21(NP))/(A23(NP)*A11(NP)-A13(NP)*      INT24820
     +                A21(NP))                                          INT24830
      DELF(NP) = (E1-A13(NP)*DELV(NP))/A11(NP)                          INT24840
      J     = NP                                                        INT24850
600   J     = J-1                                                       INT24860
      E3    = R3(J)-DELU(J+1)+A(J+1)*DELV(J+1)                          INT24870
      DEN2  = A21(J)*A12(J)*A(J+1)-A21(J)*A13(J)-A(J+1)*A22(J)*A11(J)+  INT24880
     +             A23(J)*A11(J)                                        INT24890
      DELV(J)  = (A11(J)*(R2(J)+E3*A22(J))-A21(J)*R1(J)-E3*A21(J)*A12(J)INT24900
     +               )/DEN2                                             INT24910
      DELU(J)  =-A(J+1)*DELV(J)-E3                                      INT24920
      DELF(J)  = (R1(J)-A12(J)*DELU(J)-A13(J)*DELV(J))/A11(J)           INT24930
      IF(J .GT. 1) GO TO 600                                            INT24940
C                                                                       INT24950
      DO 700 J=1,NP                                                     INT24960
      F(J,2)= F(J,2)+DELF(J)                                            INT24970
      U(J,2)= U(J,2)+DELU(J)                                            INT24980
      V(J,2)= V(J,2)+DELV(J)                                            INT24990
700   CONTINUE                                                          INT25000
      U(1,2)= 0.0                                                       INT25010
      CALL EDGCHK(NP,ETA,F(1,2),U(1,2),V(1,2))                          INT25020
      RETURN                                                            INT25030
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT25040
```

```
      END                                                               INT25050
C          DATA SET KCBCSOLV4  AT LEVEL 001 AS OF 08/24/84              INT25060
C          DATA SET KCBCSOLV4  AT LEVEL 001 AS OF 08/24/84              INT25070
C          DATA SET KCBCSOLV4  AT LEVEL 001 AS OF 02/21/84              INT25080
      SUBROUTINE SOLV4(GAMMA1,GAMMA2)                                   INT25090
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT25100
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT25110
      COMMON /BLC2/ DELF(101),DELU(101),DELV(101),DELW(101)             INT25120
      COMMON /BLC6/ S1(101),S2(101),S3(101),S4(101),S5(101),S6(101),    INT25130
     +              S7(101),S8(101),R1(101),R2(101),R3(101),R4(101)     INT25140
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT25150
      COMMON /BLCB/ A11(101),A12(101),A13(101),A14(101),                INT25160
     +              A21(101),A22(101),A23(101),A24(101)                 INT25170
C                                                                       INT25180
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT25190
C                                                                       INT25200
      A11(1)   = 1.0                                                    INT25210
      A12(1)   = 0.0                                                    INT25220
      A13(1)   = 0.0                                                    INT25230
      A14(1)   = 0.0                                                    INT25240
      A21(1)   = 0.0                                                    INT25250
      A22(1)   = 1.0                                                    INT25260
      A23(1)   = 0.0                                                    INT25270
      A24(1)   = 0.0                                                    INT25280
      DO 10 J = 2,NP                                                    INT25290
      AA1      = A13(J-1)-A(J)*A12(J-1)                                 INT25300
      AA2      = A23(J-1)-A(J)*A22(J-1)                                 INT25310
      AA3      = S2(J)-A(J)*S6(J)                                       INT25320
      DET      = AA2*A11(J-1)-AA1*A21(J-1)                              INT25330
      AJS      = A(J)**2                                                INT25340
      G11   = -(AA2+A21(J-1)*AJS)/DET                                   INT25350
      G12   = (A11(J-1)*AJS+AA1)/DET                                    INT25360
      G13   = A12(J-1)*G11+A22(J-1)*G12+A(J)                            INT25370
      G14   = A14(J-1)*G11+A24(J-1)*G12                                 INT25380
      G21   = (S4(J)*AA2-A21(J-1)*AA3)/DET                              INT25390
      G22   = (A11(J-1)*AA3-S4(J)*AA1)/DET                              INT25400
      G23   = A12(J-1)*G21+A22(J-1)*G22-S6(J)                           INT25410
      G24   = A14(J-1)*G21+A24(J-1)*G22-S8(J)                           INT25420
      A11(J)   = 1.0                                                    INT25430
      A12(J)   = -A(J)-G13                                              INT25440
      A13(J)   = A(J)*G13                                               INT25450
      A14(J)   = -G14                                                   INT25460
      A21(J)   = S3(J)                                                  INT25470
      A22(J)   = S5(J)-G23                                              INT25480
      A23(J)   = S1(J)+A(J)*G23                                         INT25490
      A24(J)   = S7(J)-G24                                              INT25500
      R1(J)    = R1(J) -G11*R1(J-1)-G12*R2(J-1)-R3(J-1)*G13             INT25510
     +                 -G14*R4(J-1)                                   INT25520
      R2(J)    = R2(J) -G21*R1(J-1)-G22*R2(J-1)-R3(J-1)*G23             INT25530
     +                 -G24*R4(J-1)                                   INT25540
10    CONTINUE                                                          INT25550
C                                                                       INT25560
C   BACKWARD SWEEP                                                      INT25570
      J        = NP                                                     INT25580
      G1       = GAMMA1/GAMMA2                                          INT25590
      R3(J)    = R3(J)/GAMMA2                                           INT25600
```

```
      R1(J)    = R1(J)-A12(J)*(R4(J)+R3(J))-A14(J)*R3(J)                INT25610
      R2(J)    = R2(J)-A22(J)*(R4(J)+R3(J))-A24(J)*R3(J)                INT25620
      C1       = A11(J)-G1*(A12(J)+A14(J))                              INT25630
      C2       = A21(J)-G1*(A22(J)+A24(J))                              INT25640
      DET      = C1*A23(J)-C2*A13(J)                                    INT25650
      DELF(J)  = (R1(J)*A23(J)-R2(J)*A13(J))/DET                        INT25660
      DELV(J)  = (C1*R2(J)-C2*R1(J))/DET                                INT25670
      DELW(J)  = R3(J)-G1*DELF(J)                                       INT25680
      DELU(J)  = R4(J)+DELW(J)                                          INT25690
20    J        = J-1                                                    INT25700
      CC1      = DELU(J+1)-R3(J)-A(J+1)*DELV(J+1)                       INT25710
      CC2      = DELW(J+1)-R4(J)                                        INT25720
      CC3      = A13(J)-A(J+1)*A12(J)                                   INT25730
      CC4      = R1(J)-A12(J)*CC1-A14(J)*CC2                            INT25740
      CC5      = A23(J)-A(J+1)*A22(J)                                   INT25750
      CC6      = R2(J)-A22(J)*CC1-A24(J)*CC2                            INT25760
      DENO     = A11(J)*CC5-A21(J)*CC3                                  INT25770
      DELF(J)  = (CC4*CC5-CC3*CC6)/DENO                                 INT25780
      DELV(J)  = (A11(J)*CC6-A21(J)*CC4)/DENO                           INT25790
      DELW(J)  = CC2                                                    INT25800
      DELU(J)  = CC1-A(J+1)*DELV(J)                                     INT25810
      IF(J .GE. 2) GO TO 20                                             INT25820
      DO 30 J = 1,NP                                                    INT25830
      F(J,2) = F(J,2)+DELF(J)                                           INT25840
      U(J,2) = U(J,2)+DELU(J)                                           INT25850
      V(J,2) = V(J,2)+DELV(J)                                           INT25860
      W(J,2) = W(J,2)+DELW(J)                                           INT25870
30    CONTINUE                                                          INT25880
      U(1,2) = 0.0                                                      INT25890
      CALL EDGCHK(NP,ETA,F(1,2),U(1,2),V(1,2))                          INT25900
      RETURN                                                            INT25910
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INT25920
      END                                                               INT25930
C          DATA SET KCBCTRNS   AT LEVEL 001 AS OF 08/24/84              INT25940
C          DATA SET KCBCTRNS   AT LEVEL 001 AS OF 08/24/84              INT25950
C          DATA SET KCBCTRNS   AT LEVEL 005 AS OF 03/13/84              INT25960
      SUBROUTINE TRNS(ICODE)                                            INT25970
C                                                                       INT25980
C   CALCULATE TRANSITION LOCATION USING MICHEL CRITERION                INT25990
C                                                                       INT26000
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT26010
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT26020
      COMMON/BLIN/ TITLE(20),XC(100),YC(100),ISG(100),DELS(100),        INT26030
     +             XCTR,XTR,ISTRP,ICYCLE,ICYTL,XCTRS(2),TRFIND(2)       INT26040
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                    INT26050
     +                    ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT      INT26060
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT26070
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT26080
      COMMON/TRN/ PGAMTR,OMEGA,RTHETB,RTRANB                            INT26090
C   ------------------------------------------------------------        INT26100
100   FORMAT(/3X,'BEGIN OF TRANSITION HAS BEEN DETECTED BY MICHEL''S ', INT26110
     +'CRITERION: X/C =',F8.4,4X,'S/C =',F8.4,4X,'NTR =',I3/)           INT26120
110   FORMAT(/3X,'BEGIN OF TRANSITION IS ASSUMED AT THE POINT OF ',     INT26130
     +'LAMINAR SEPARATION: X/C =',F8.4,4X,'S/C =',F8.4,4X,'NTR =',I3/)  INT26140
C   ------------------------------------------------------------        INT26150
      ICODE   = 0                                                       INT26160
```

```
      ISEP    = 1                                                    INT26170
C                                                                    INT26180
      IF(V(1,2).LT. 0.0) THEN                                        INT26190
C *** TRANSITION PROCESS HAS BEGUN DUE TO LAMINAR SEPARATION ***     INT26200
         FAC = V(1,1)/(V(1,1)-V(1,2))                                INT26210
         GOTO 20                                                     INT26220
      END IF                                                         INT26230
C                                                                    INT26240
C *** CHECK MICHEL'S TRANSITION CRITERION ***                        INT26250
      ISEP   = 0                                                     INT26260
      SUM    = 0.0                                                   INT26270
      F1     = U(1,2)/U(NP,2)*(1.0-U(1,2)/U(NP,2))                   INT26280
      DO 10 J=2,NP                                                   INT26290
      F2     = U(J,2)/U(NP,2)*(1.0-U(J,2)/U(NP,2))                   INT26300
      SUM    = SUM + (F1 + F2 ) *A(J)                                INT26310
      F1     = F2                                                    INT26320
   10 CONTINUE                                                       INT26330
      CONV   = SQRT(RL/X(NX))                                        INT26340
      IF(NX.LE.NS) CONV = SQRT(RX)/X(NX)                             INT26350
      THETA  = SUM / CONV                                            INT26360
      RTHETA = RL * UE(NX) * THETA                                   INT26370
      RTRAN  = 1.174 * (1.0+22400.0/RX) * RX**0.46                   INT26380
      IF(RTHETA.LT.RTRAN) THEN                                       INT26390
         RTHETB = RTHETA                                             INT26400
         RTRANB = RTRAN                                              INT26410
         RETURN                                                      INT26420
      END IF                                                         INT26430
C                                                                    INT26440
C *** TRANSITION PROCESS HAS BEGUN BECAUSE OF MICHEL'S CRITERION *** INT26450
      FAC = (RTHETB-RTRANB)/(RTRAN-RTRANB-RTHETA+RTHETB)             INT26460
C                                                                    INT26470
C *** COMPUTE EXACT LOCATION OF TRANSITION BEGIN ***                 INT26480
   20 NTR  = NX-1                                                    INT26490
      NTR1 = NTR + 1                                                 INT26500
      XCTR = XC(NX-1) + FAC*(XC(NX)-XC(NX-1))                        INT26510
      XTR  = X(NX-1) + FAC*(X(NX)-X(NX-1))                           INT26520
      UETR = UE(NX-1) + FAC*(UE(NX)-UE(NX-1))                        INT26530
      IF ( ISEP .EQ. 0 ) WRITE (6,100) XCTR,XTR,NTR                  INT26540
      IF ( ISEP .EQ. 1 ) WRITE (6,110) XCTR,XTR,NTR                  INT26550
      ICODE  = 1                                                     INT26560
C                                                                    INT26570
C *** CALCULATE INTERMITTENCY DISTRIBUTION ***                       INT26580
      RXNTR  = XTR * UETR * RL                                       INT26590
      GGFT   = RL**2/PGAMTR/RXNTR**1.34*UETR**3                      INT26600
      DO 30 I=NTR1,NXT                                               INT26610
      ALFAS(I) = 0.0168                                              INT26620
      GMTRS(I)= 1.0                                                  INT26630
   30 CONTINUE                                                       INT26640
      ALFAS(NTR) = 0.0168                                            INT26650
      UEINTG = 0.0                                                   INT26660
      U1 = 0.5/UETR                                                  INT26670
      X1 = XTR                                                       INT26680
      DO 40 I=NTR1,NXT                                               INT26690
      U2 = 0.5/UE(I)                                                 INT26700
      X2 = X(I)                                                      INT26710
      UEINTG = UEINTG+(U1+U2)*(X2-X1)                                INT26720
```

```
      U1 = U2
      X1 = X2
      GG = GGFT*UEINTG*(X(I)-XTR)
      IF(GG .GT. 10.0) GOTO 50
      GMTRS(I) = 1.0-EXP(-GG)
   40 CONTINUE
C
C *** RESET FINITE DIFFERENCE CALCULATIONS ***
   50 DO 60 J=1,NPT
      F(J,2) = F(J,1)
      U(J,2) = U(J,1)
      V(J,2) = V(J,1)
      B(J,2) = B(J,1)
      W(J,2) = W(J,1)
60    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE EDGCHK(NP, ETA, F, U, V)
C
      DIMENSION ETA(101), F(101), U(101), V(101)
C    -------------------------------------------------------
      JS    = NP - 3
      NPM1  = NP - 1
      DO 10 J=JS, NPM1
      JJ    = J
      IF(U(J).GE.U(NP) .OR. V(J).LT.0.0) GOTO 20
10    CONTINUE
      RETURN
20    JS = JJ - 1
      IF(JS.GT.(NP-2)) JS = NP-2
      CALL AMEAN(JS, NP, ETA, U, 1)
      CALL AMEAN(JS, NP, ETA, F, 1)
      DETAP  = ETA(JS) -ETA(JS-1)
      VJP    = (U(JS)-U(JS-1))/DETAP
      DO 30 J=JS,NPM1
      DETAM  = ETA(J+1)-ETA(J)
      VJM    = (U(J+1)-U(J))/DETAM
      V(J)   = (VJM*DETAP + VJP*DETAM)/(DETAP+DETAM)
      VJP    = VJM
      DETAP  = DETAM
30    CONTINUE
      V(NP)  = -V(NP-1) + 2.0 * VJP
      RETURN
C   ******************************************************************
C   NOTES:          (FOR CHANGING FROM THE ORIGINAL PROGRAM)          *
C                                                                     *
C   1.  'EDDY' HAS BEEN MODIFIED BY ADDING 'FINT'.                    *
C   2.  SUBROUTINE 'EDGCHK' HAS BEEN ADDED.                           *
C   3. GROWTH LIMIT HAS BEEN ADDED FOR 2 IN 'MAIN2'.                  *
C                                                                     *
C                                                                     *
C   ******************************************************************
      END
```

```
      SUBROUTINE SMPSON                                                 INT27280
C                                                                       INT27290
      COMMON /BLC0/ NX,NXT,NP,NPT,NTR,IT,INVRS,NS,IP                    INT27300
      COMMON /BLC1/ F(101,2),U(101,2),V(101,2),W(101,2),B(101,2)        INT27310
      COMMON /BLC7/ C(100,100),D(100),DB(100),DBP(100),UE0(100),GI      INT27320
      COMMON/EDDY1/ RL,RX,SQRX,RXNTR,GMTR,GMTRS(100)                    INT27330
     +                    ,ALFAS(100),FFS(100),RTS(100),IEDY,NXSPT     INT27340
      COMMON /GTY / X(101),UE(100),P1(100),P2(100),CEL,CELH             INT27350
      COMMON /GRD / ETA(101),DETA(101),A(101)                           INT27360
      DIMENSION CRD(12),RTD(12)                                         INT27370
      DATA RTD/0.00,0.05,0.12,0.20,0.30,0.40,0.50,0.60,0.70,            INT27380
     +             0.80,0.90,1.00/                                      INT27390
      DATA CRD/5.00,4.75,4.35,3.80,3.25,2.85,2.58,2.37,2.25,            INT27400
     +             2.15,2.06,2.00/                                      INT27410
C                                                                       INT27420
C   ------------------------------------------------------------        INT27430
C                                                                       INT27440
C   STEP 1 CALCULATE (DU/DX)/(DU/DY)                                    INT27450
C     IF(NX.LT.NXSPT) GOTO 10                                           INT27460
C                                                                       INT27470
C   IN THE SEPARATED REGION, ALFA SET TO BE CONSTANT                    INT27480
C     ALFAS(NX)= ALFASP                                                 INT27490
C     RETURN                                                            INT27500
C(-----                                                                 INT27510
C                                                                       INT27520
C  10 CONTINUE                                                          INT27530
C     IF(V(1,2).GT. 0.0) GOTO 20                                        INT27540
C                                                                       INT27550
C   SEPARATION OCCURS. ALFA SET TO BE THE PREVIOUS ITERATED VALUE       INT27560
C     ALFASP = ALFAS(NX)                                                INT27570
C     NXSPT  = NX                                                       INT27580
C     RETURN                                                            INT27590
C(-----                                                                 INT27600
C   MODIFY OUTER EDDY BASED ON SIMPSON SUGGESTION                       INT27610
      TM     = 0.0                                                      INT27620
      JM     = 1                                                        INT27630
      DO 30 J=2,NP                                                      INT27640
      TS     = (B(J,2)-1.0)* V(J,2)                                     INT27650
      IF(TS.LT.TM) GOTO 30                                              INT27660
      TM     = TS                                                       INT27670
      JM     = J                                                        INT27680
30    CONTINUE                                                          INT27690
      VNXM = 0.5*(V(JM,2)+V(JM,1))                                      INT27700
      IF (NX .LE. NS) GOTO 35                                           INT27710
      DUDX = (U(JM,2)-U(JM,1)) / (X(NX)-X(NX-1))                        INT27720
      GO TO 38                                                          INT27730
35    DUDX   = CEL*(U(JM,2)-U(JM,1))+P2(NX)*U(JM,2)+0.5*ETA(JM)*        INT27740
     +               VNXM*(P2(NX)-1.0)                                  INT27750
38    RU     = RL                                                       INT27760
      IF(NX.LE.NS)RU = RL * UE0(NX) * X(NX)                             INT27770
      RL2    = SQRT(RU)                                                 INT27780
      RR     = DUDX/VNXM/RL2                                            INT27790
C                                                                       INT27800
C   STEP 2 : CALCULATE (UU - VV)/UV                                     INT27810
      VNXM = 0.5*(V(1,2)+V(1,1))                                        INT27820
```

```
      RT    = VNXM/TM                                                    INT27830
C    PRINT'(3X,2I5,3F10.3)',NX,JM,VNXM,TM,RT                             INT27840
      IF (RT .LT. 0.0) RT = 0.0                                          INT27850
      IF(RT.GT.1.0) GOTO 60                                              INT27860
      CR    = 6.0 /(1.0 +  2.0 * RT*(2.0 -RT))                          INT27870
C     CR    = 2.0                                                        INT27880
C     DO 40 I=2,12                                                       INT27890
C     IF(RT.LT.RTD(I)) GOTO 50                                           INT27900
C  40 CONTINUE                                                           INT27910
C     GOTO 70                                                            INT27920
C  50 CR    = CRD(I-1)+(CRD(I)-CRD(I-1))*(RT-RTD(I-1))/(RTD(I)-RTD(I-1))INT27930
      GOTO 70                                                            INT27940
60    CR    = (1.0 + RT ) /RT                                            INT27950
C                                                                        INT27960
C    STEP 3 : CALCULATE FF                                               INT27970
70    FR    = CR * RR                                                    INT27980
      IF(FR .GT.0.35) FR = 0.35                                          INT27990
      IF (FR .LT. -0.8) FR = -0.8                                        INT28000
      FFS(NX)= (FFS(NX) +  (1.0 -FR))/ 2.0                               INT28010
      RTS(NX)= RT                                                        INT28020
      ALFAS(NX)= 0.0168/FFS(NX)**2.5                                     INT28030
      RETURN                                                             INT28040
C(-----                                                                  INT28050
      END                                                                INT28060
C                                                                        INT28070
      SUBROUTINE XSPACE(NI,NRITE,XII,XLLT,RAD,NL1,NR1)                   INT28080
      DIMENSION XII(200),T(200)                                          INT28090
      DATA PI/3.14159265359879/                                          INT28100
      RAD = PI                                                           INT28110
      NLEFT=NI-1-NRITE                                                   INT28120
      NR4=NRITE/2                                                        INT28130
      IF((NRITE/2*2) .NE. NRITE) NR4=(NRITE+1)/2                         INT28140
      NL1=NR4+1                                                          INT28150
      NL2=NR4+NLEFT                                                      INT28160
      NR1=NL2+1                                                          INT28170
      NR2=NI                                                             INT28180
      PI2=0.5*PI                                                         INT28190
      RAD2=(PI-RAD)/2.0+PI2                                              INT28200
      RAD3=RAD2+RAD                                                      INT28210
      SRT =RAD2/FLOAT(NR4)                                               INT28220
      SRT2=SRT                                                           INT28230
      IF((NRITE/2*2) .NE. NRITE) SRT2=RAD2/FLOAT(NR4-1)                  INT28240
      SLT = RAD/FLOAT(NLEFT)                                             INT28250
      DO 10 I=1,NR4                                                      INT28260
10    XII(I)=0.5*(1.0+COS(FLOAT(I-1)*SRT))                               INT28270
      DO 20 I=NL1,NL2                                                    INT28280
20    XII(I)=0.5+XLLT*COS(FLOAT(I-NL1)*SLT+RAD2)                         INT28290
      DO 30 I=NR1,NR2                                                    INT28300
30    XII(I)=0.5*(1.0+COS(FLOAT(I-NR1)*SRT2+RAD3))                       INT28310
      NA=(NI+1)/2                                                        INT28320
      IF((NI/2*2) .EQ. NI) NA=NI/2+1                                     INT28330
      FN1=FLOAT(NA-1)                                                    INT28340
      FN2=FN1                                                            INT28350
      IF((NI/2*2) .EQ. NI) FN2=FLOAT(NA-2)                               INT28360
      DO 40 I=1,NA                                                       INT28370
40    T(I)=FLOAT(NA-I)/FN1                                               INT28380
```

```
      CALL AMEAN(1,NA,T,XII,1)                                          INT28390
      XDIF = XII(1) - XII(2)                                            INT28400
      IF(XDIF .LT. 0.004) THEN                                          INT28410
      DO 45 I=2,5                                                       INT28420
      XII(I) = XII(I-1)-XDIF*3.0                                        INT28430
45    CONTINUE                                                          INT28440
      CALL AMEAN(2,NA,T,XII,10)                                         INT28450
      END IF                                                            INT28460
      DO 50 I=NA,NI                                                     INT28470
50    XII(I) = XII(NI-I+1)                                              INT28480
      RETURN                                                            INT28490
      END                                                               INT28500
      SUBROUTINE TRGRID (N1 , XO , YO,NI,NRITE,XLLT,N1O,RAD,ID,NXSS)    INT28510
C  THIS SUB. IS TO REGRID SPACING NEAR TRAILING-EDGE                    INT28520
C                                                                       INT28530
      DIMENSION XO(200),YO(200),XI(200),YI(200),D1(200),D2(200),D3(200),INT28540
     +          XOO(200),YOO(200),XII(200),YII(200),WX(200),WY(200),    INT28550
     +          WXI(200),WYI(200),T(200)                                INT28560
C                                                                       INT28570
      N2O    =  N1O                                                     INT28580
      IF((N1/2*2) .EQ. N1) N2O = N1O+1                                  INT28590
      IF((NI-(NI/2)*2) .NE. 0) N1I= (NI-1)/2+1                          INT28600
      IF((NI-(NI/2)*2) .EQ. 0) N1I=NI/2                                 INT28610
      N2I    =  N1I                                                     INT28620
      IF((NI/2*2) .EQ. NI) N2I = N1I+1                                  INT28630
C                                                                       INT28640
      CALL XSPACE(NI,NRITE,XI,XLLT,RAD,NL1,NR1)                         INT28650
C      PRINT *,'NRITE=',NRITE,' XLLT=',XLLT                             INT28660
C     WRITE (6, 290)                                                    INT28670
C     WRITE (6, 300) (XO(I) ,I=1,N1)                                    INT28680
C     WRITE (6, 298)                                                    INT28690
C     WRITE (6, 300) (YO(I) ,I=1,N1)                                    INT28700
      IF(ID .EQ. 2) THEN                                                INT28710
      DO 60 I=NL1,NXSS                                                  INT28720
      YI(I)=YO(I)                                                       INT28730
60    XI(I)=XO(I)                                                       INT28740
      NXST=NXSS+8                                                       INT28750
      XM1=(XI(NXST)-XI(NXSS))/8.0                                       INT28760
      XM2=(XI(NR1)-XI(NXSS))/(NR1-NXSS)                                 INT28770
      DO 62 I=NXSS,NR1-1                                                INT28780
62    XI(I)=(XO(I)+XI(I))/2.0                                           INT28790
      DO 65 I=NXSS,NXST                                                 INT28800
65    T(I)=FLOAT(I-NXSS)*XM1+XI(NXSS)                                   INT28810
      CALL AMEAN(NXSS,NXST,T,XI,8)                                      INT28820
      DO 68 I=NXSS,NR1                                                  INT28830
68    T(I)=FLOAT(I-NXSS)*XM2+XI(NXSS)                                   INT28840
      CALL AMEAN(NXSS,NR1,T,XI,12)                                      INT28850
      N1O=NL1                                                           INT28860
      N1I=N1O                                                           INT28870
      N2O=N1+1-NXSS                                                     INT28880
      N2I=NI-NXSS+1                                                     INT28890
      ELSE                                                              INT28900
      NXSS=N2I                                                          INT28910
      END IF                                                            INT28920
C  FOR LOWER SURFACE                                                    INT28930
      DO 5 I = 1 , N1O                                                  INT28940
```

```
      II = N1O - I + 1                                                  INT28950
      WX(II) = XO(I)                                                    INT28960
      WY(II) = YO(I)                                                    INT28970
5     CONTINUE                                                          INT28980
      DO 7 I = 1 , N1I                                                  INT28990
      II = N1I - I + 1                                                  INT29000
      WXI(II) = XI(I)                                                   INT29010
7     CONTINUE                                                          INT29020
      CALL DIFF3(N1O,WX,WY,D1,D2,D3,0)                                  INT29030
      CALL INTRP3(N1O,WX,WY,D1,D2,D3,N1I,WXI,WYI)                       INT29040
      DO 9 I = 1 , N1I                                                  INT29050
      II = N1I - I + 1                                                  INT29060
      YI(II) = WYI(I)                                                   INT29070
9     CONTINUE                                                          INT29080
C                                                                       INT29090
C  FOR UPPER SURFACE                                                    INT29100
      DO 10 I = 1 , N2O                                                 INT29110
      II = N1 - N2O + I                                                 INT29120
      XOO(I) = XO(II)                                                   INT29130
      YOO(I) = YO(II)                                                   INT29140
10    CONTINUE                                                          INT29150
      DO 20 I = 1 , N2I                                                 INT29160
      II = NI - N2I + I                                                 INT29170
      WXI(I) = XI(II)                                                   INT29180
20    CONTINUE                                                          INT29190
      CALL DIFF3(N2O,XOO,YOO,D1,D2,D3,0)                                INT29200
      CALL INTRP3(N2O,XOO,YOO,D1,D2,D3,N2I,WXI,WYI)                     INT29210
      DO 25 I = 1 , N2I                                                 INT29220
      II = N2I - I + 1                                                  INT29230
      XII(II) = WXI(I)                                                  INT29240
      YII(II) = WYI(I)                                                  INT29250
25    CONTINUE                                                          INT29260
C                                                                       INT29270
C  COMBINE TWO SURFACES INTO ONE CIRCLE                                 INT29280
C                                                                       INT29290
C     NN = N1 - N1O - N2O                                               INT29300
C     DO 30 I = 1 , NN                                                  INT29310
C     II = N1I + I                                                      INT29320
C     III= N1O + I                                                      INT29330
C     XI(II) = XO(III)                                                  INT29340
C     YI(II) = YO(III)                                                  INT29350
C30   CONTINUE                                                          INT29360
      DO 40 I = 1 , N2I                                                 INT29370
      I1 = NXSS + I -1                                                  INT29380
      I2 = N2I - I + 1                                                  INT29390
      XI(I1) = XII(I2)                                                  INT29400
      YI(I1) = YII(I2)                                                  INT29410
40    CONTINUE                                                          INT29420
      XI(1) = XO(1)                                                     INT29430
      XI(I1)= XO(N1)                                                    INT29440
      YI(1) = YO(1)                                                     INT29450
      YI(I1)= YO(N1)                                                    INT29460
C                                                                       INT29470
      N1 = I1                                                           INT29480
      DO 50 I = 1 , N1                                                  INT29490
      XO(I) = XI(I)                                                     INT29500
```

```
      YO(I) = YI(I)                                                    INT29510
50    CONTINUE                                                         INT29520
C                                                                      INT29530
C     WRITE (6 , 295)                                                  INT29540
C     WRITE (6 , 300) (XO(I) , I=1,N1)                                 INT29550
C     WRITE (6 , 298)                                                  INT29560
C     WRITE (6 , 300) (YO(I) , I=1,N1)                                 INT29570
C                                                                      INT29580
      RETURN                                                           INT29590
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -    INT29600
100   FORMAT(7I5)                                                      INT29610
200   FORMAT(6F10.0)                                                   INT29620
290   FORMAT(/,'  ORIGINAL COORDINATES',/,'  X/C')                     INT29630
295   FORMAT(/,'  INTERPLATED COORDINATES',/,'  X/C')                  INT29640
298   FORMAT('      Y/C')                                              INT29650
300   FORMAT(6F10.6)                                                   INT29660
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -    INT29670
      END                                                              INT29680
C                                                                      INT29690
      SUBROUTINE STAGR(N,STAG,XO,YO,XSTGR,YSTGR)                       INT29700
C                                                                      INT29710
      DIMENSION XO(100),YO(100),XSTGR(100),YSTGR(100),DS(100)          INT29720
C                                                                      INT29730
      XOTE = 0.5 * (XO(1)+XO(N))                                       INT29740
      YOTE = 0.5 * (YO(1)+YO(N))                                       INT29750
      DS(1) = SQRT((XO(1)-XOTE)**2 + (YO(1)-YOTE)**2)                  INT29760
      DSM = DS(1)                                                      INT29770
      DO 10 I = 2 , N                                                  INT29780
      DS(I) = SQRT((XO(I)-XOTE)**2 + (YO(I)-YOTE)**2)                  INT29790
      IF (DS(I) .LT. DSM) GOTO 10                                      INT29800
      IM = I                                                           INT29810
      DSM = DS(I)                                                      INT29820
10    CONTINUE                                                         INT29830
C                                                                      INT29840
      YYY = YOTE-YO(IM)                                                INT29850
      XXX = XOTE-XO(IM)                                                INT29860
      IF (YYY .EQ. 0.0 .AND. XXX .EQ. 0.0) THEN                        INT29870
      ANG = 0.0                                                        INT29880
      ELSE                                                             INT29890
      ANG = ATAN2(YYY,XXX)                                             INT29900
      END IF                                                           INT29910
      ANG = ANG + STAG                                                 INT29920
C                                                                      INT29930
      COSAN = COS(ANG)                                                 INT29940
      SINAN = SIN(ANG)                                                 INT29950
      DO 20 I = 1 , N                                                  INT29960
C     YY = YO(I)-YO(IM)                                                INT29970
C     XX = XO(I)-XO(IM)                                                INT29980
C     IF (YY .EQ. 0.0 .AND. XX .EQ. 0.0) THEN                          INT29990
C     ANGCO = 0.0                                                      INT30000
C     ELSE                                                             INT30010
C     ANGCO = ATAN2(YY,XX)                                             INT30020
C     END IF                                                           INT30030
      XSTGR(I)= XO(I)*COSAN + YO(I)*SINAN                              INT30040
      YSTGR(I)= YO(I)*COSAN - XO(I)*SINAN                              INT30050
20    CONTINUE                                                         INT30060
```

```
      RETURN                                                        INT30070
      END                                                           INT30080
```

```
      YU(I) = YC + COS(DY)*YT                                          C4 00220
      YL(I) = YC - COS(DY)*YT                                          C4 00230
      XU(I) = X(I) - SIN(DY)*YT                                        C4 00240
      XL(I) = X(I) + SIN(DY)*YT                                        C4 00250
  100 CONTINUE                                                         C4 00260
C     WRITE (6,900) (I,X(I),XU(I),YU(I),XL(I),YL(I),I=0,NMAX)          C4 00270
C 900 FORMAT (I5,4X,F10.6,4X,2F10.6,4X,2F10.6)                         C4 00280
      WRITE (1,910) (XL(I),I=NMAX,0,-1),(XU(I),I=1,NMAX)               C4 00290
      WRITE (1,910) (YL(I),I=NMAX,0,-1),(YU(I),I=1,NMAX)               C4 00300
  910 FORMAT (6F10.6)                                                  C4 00310
      STOP                                                             C4 00320
      END                                                              C4 00330
```

LIST OF REFERENCES

1. Cebeci, T., Clark, R. W., Chang, K. C., Halsey, N. D. and Lee, K., *Airfoils with Separation and the Resulting Wake,* Journal of Fluid Mechanics, Vol 163, pp. 323-347, 1986.

2. Cebeci, T. and Bradshaw, P., *Momentum Transfer in Boundary Layers,* McGraw-Hill Book Company, New York, 1977.

3. Krainer, A., *Viscous Inviscid Interaction Analysis of Incompressible Cascade Flows,* NPS-67-86-005 CR, Naval Postgraduate School, Monterey, CA. December 1986.

4. Schlichting, H., *Boundary Layer Theory,* McGraw-Hill Book Company, New York, 1968.

5. Lighthill, M. J., *On Displacement Thickness,* Journal of Fluid Mechanics, Vol 4, 1958.

6. Cebeci, T. and Smith, A.M.O., *Analysis of Turbulent Boundary Layers,* Academic Press, New York, 1974.

7. Rodi, W. and Schonung, B., *Interaktives Inverses Grenzschichtverfahren zur Berechnung von lokalen Ablöseblasen an Turbinenschaufeln,* Z. Flugwiss Weltraumforsch, Vol. 11, pp. 271-280, 1987.

8. Elazar, Y., *A Mapping of the Viscous Flow Behavior in a Controlled Diffusion Compressor Cascade Using Laser Doppler Velocimeter and Preliminary Evaluation of Codes for the Prediction of Stall,* Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA. March 1988.

9. Hobbs, Wagner, Donnenhoffer and Dring, *Supercritical Airfoil Technology Program,* United Technologies Corporation, Pratt & Whitney Aircraft Group, West Palm Beach, FL. contract N00019-79-C-0229, September 1980.

10. Walker, G. J., *The Turbulent Boundary Layer on an Axial Compressor Blade,* The American Society of Mechnical Engineers, ASME paper 82-GT-52, 1982.

11. Deutch, S. and Zierk, W. C., *The Measurement of Boundary Layers on a Compressor Blade in Cascade,* Journal of Turbomachinery, Vol. 109, October 1987, pp. 520-526.

12. Hoheisel, H. and Seyb, N. J., *The Boundary Layer Behavior of Highly Loaded Compressor Cascade At Transonic Flow Conditions,* North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, AGARD-CPP-400 401, September 1986.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Department of Aeronautics, Code 67 Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor T. Cebeci Staff director Research & Technology McDonnell Douglas Corporation 3855 Lakewood Boulevard Long Beach, CA 90846	1
5.	Professor M. F. Platzer Department of Aeronautics, Code 67Pl Naval Postgraduate School Monterey, CA 93943-5000	12
6.	Mr. T. S. Momiyama Director of Aircraft Division Naval Air System Command, Code 931 Washington D.C. 20361	1
7.	Mr. G. Derderian Project Manager for Propulsion, Aircraft Division Naval Air System Command, Code 931 Washington D.C. 20361	1
8.	Professor G. J. Walker Department of Civil and Mechaical Engineering University of Tasmania Hobart, Tasmania, Australia	1
9.	Professor R. P. Shreeve Turbopropulsion Laboratory, Code 67Sf Naval Postgraduate School Monterey, CA 93943-5000	1

10. Professor D. W. Netzer 1
Department of Aeronautics, Code 67Nt
Naval Postgraduate School
Monterey, CA 93943-5000

11. Dr. Stephen J. Shamroth 1
Scientific Research Association, inc.
50 Nye Road, P.O. Box 1058
Clastonbury, CO 06033

12. Dr. Robert J. Simoneau 1
Chief of Heat Transfer Branch, Mail Stop 5-11
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

13. Mr. James O'Brien 1
Heat Transfer Branch, Mail Stop 5-11
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

14. Moshe Horev 1
Israel Defense Force
I.D.F., P.O. Box 02158
Israel

15. Defense and Armed Forces Attache 1
Embassy of Israel
3514 International Drive, N.W.
Washington, D.C. 20008

16. Isaak Mostov 1
SMC 2752
Naval Postgraduate School
Monterey, CA 93943-5000

17. Zeev Snir 1
Israel Defense Force
I.D.F., P.O. Box 02158
Israel

Thesis
S65922 Snir
c.1 Investigation of incompressible cascade flows using a viscous/inviscid interactive code.

Thesis
S65922 Snir
c.1 Investigation of incompressible cascade flows using a viscous/inviscid interactive code.

AMERICAN BINDERIES

thesS65922
Investigation of incompressible cascade
3 2768 000 81552 6
DUDLEY KNOX LIBRARY